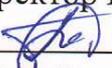
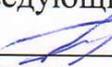


МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

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

Кафедра «Государственного, муниципального и корпоративного управления»

«СОГЛАСОВАНО» Директор ИМА  Бодров О.А. « 26 » 06 2020 г.	 «УТВЕРЖДАЮ» Проректор по РОПиМД Корячко А.В. « 26 » 06 2020 г.
Заведующий кафедрой  Перфильев С.В. « 26 » 06 2020 г.	

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Б1.В.ДВ.02.01 «Современные технологии баз данных»

Направление подготовки

38.04.04 «Государственное и муниципальное управление»

Профиль – Информационные технологии в государственном и муниципальном
управлении

ОПОП академической магистратуры

«Государственное и муниципальное управление»

Формы обучения – очно-заочная

Рязань, 2020

ЛИСТ СОГЛАСОВАНИЙ

Рабочая программа составлена с учетом требований федерального государственного образовательного стандарта высшего образования по направлению подготовки 38.04.04 «Государственное и муниципальное управление», утвержденным приказом Минобрнауки России от 26 ноября 2014 г. N 1518

Разработчики

к.т.н., доцент кафедры ЭВМ

(должность, кафедра)

Грица

(подпись)

Тринченко Н.Н.

(Ф.И.О.)

Рассмотрена и утверждена на заседании кафедры «11» июня 2020 г., протокол № 10

Заведующий кафедрой

ЭВМ

(кафедра)

Костров

(подпись)

Костров Б.В.

(Ф.И.О.)

1. ПЕРЕЧЕНЬ ПЛАНИРУЕМЫХ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ, СООТНЕСЕННЫХ С ПЛАНИРУЕМЫМИ РЕЗУЛЬТАТАМИ ОСВОЕНИЯ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

Программа по дисциплине «Современные технологии баз данных» составлена с требованиями, установленными Федеральным государственным образовательным стандартом высшего образования по направлению подготовки 38.04.04 Государственное и муниципальное управление (уровень магистратуры), утвержденным Приказом Минобрнауки России № 1518 от 26.11.2014 г. (с изм. и доп.).

Программа предназначена для студентов, обучающихся по основной профессиональной образовательной программе высшего образования (далее - ОПОП ВО) «Информационные технологии в государственном и муниципальном управлении», реализуемой в рамках направления подготовки 38.04.04 Государственное и муниципальное управление (уровень магистратуры).

Целью дисциплины «Современные технологии баз данных» является изучение современных подходов и технологий в области проектирования, программирования и сопровождения баз данных (БД) с использованием систем управления базами данных (СУБД).

Задачами дисциплины являются:

- получение теоретических знаний о современных подходах и технологиях разработки БД, принципах построения запросов к БД на языке SQL и его процедурных расширениях, принципах работы серверов БД;
- приобретение практических навыков использования автоматизированных средств разработки БД, навыков создания БД с использованием современных серверных СУБД;
- овладение принципами, стандартами и средствами применения основных технологий БД при проектировании информационных систем различного назначения.

Перечень планируемых результатов обучения по дисциплине

<i>Коды компетенции</i>	<i>Результаты освоения ООП Содержание компетенций</i>	<i>Перечень планируемых результатов обучения при прохождении практики</i>
ПК-11	способностью осуществлять верификацию и структуризацию информации, получаемой из разных источников	<p><i>знать:</i></p> <ul style="list-style-type: none"> – современные технологии проектирования и исследования моделей данных; – нотации проектирования моделей данных, используемые при разработке информационных систем для решения задач в различных предметных областях; – основные принципы и правила проверки правильности результатов проектирования моделей данных; принципы построения типовых запросов к данным, получаемых из разных источников; принципы работы современных инструментальных средств проектирования БД; <p><i>уметь:</i></p> <ul style="list-style-type: none"> – разрабатывать модели данных с использованием существующих технологий для различных предметных областей; – проводить исследование моделей данных для проверки правильности результата проектирования; – разрабатывать сценарии создания и использования основных объектов БД на различных диалектах языка SQL в современных СУБД; <p><i>владеть:</i></p> <ul style="list-style-type: none"> – навыками построения и анализа правильности реализации моделей данных при проектировании информационных систем для различных предметных областей;

<i>Коды компетенции</i>	<i>Результаты освоения ООП Содержание компетенций</i>	<i>Перечень планируемых результатов обучения при прохождении практики</i>
		<ul style="list-style-type: none"> – основными приемами проектирования моделей данных с использованием различных нотаций, разработки БД в современных СУБД; – навыками разработки типовых запросов к данным, получаемых из разных источников; программирования основных объектов БД с использованием различных диалектов языка SQL для современных СУБД.

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП

Дисциплина реализуется в рамках вариативной части Блока 1 учебного плана ОПОП, являясь дисциплиной по выбору. Дисциплина изучается на 2 курсе в 3 семестре.

Знания, полученные в ходе изучения дисциплины «Современные технологии баз данных» логически связаны с дисциплинами, изучаемыми студентами ранее, например: «Информационно-аналитические технологии государственного управления», «Экономико-математическое моделирование».

Материал дисциплины формирует информационные основы для освоения таких дисциплин как «Интеллектуально-аналитические системы принятия управленческих решений», «Логистика», а так же НИР, практик и выпускной квалификационной работы.

3. ОБЪЕМ ДИСЦИПЛИНЫ (МОДУЛЯ) В ЗАЧЕТНЫХ ЕДИНИЦАХ С УКАЗАНИЕМ КОЛИЧЕСТВА АКАДЕМИЧЕСКИХ ЧАСОВ, ВЫДЕЛЕННЫХ НА КОНТАКТНУЮ РАБОТУ ОБУЧАЮЩИХСЯ С ПРЕПОДАВАТЕЛЕМ (ПО ВИДАМ ЗАНЯТИЙ) И НА САМОСТОЯТЕЛЬНУЮ РАБОТУ ОБУЧАЮЩИХСЯ

Общая трудоемкость (объем) дисциплины составляет 5 зачетных единиц (ЗЕ), 180 часов.

Объем дисциплины	Всего часов
Общая трудоемкость дисциплины	180
1. Контактная работа обучающихся с преподавателем (всего), в том числе:	50,65
лекции	16
практические занятия	16
лабораторные работы	16
консультации	2
иные виды контактной работы	0,65
2. Самостоятельная работа обучающихся (всего), в том числе:	76
курсовой проект (работа)	11,7
самостоятельные занятия	64,3
3. Контроль	53,35
Вид промежуточной аттестации обучающихся	Экзамен, КР

4. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ), СТРУКТУРИРОВАННОЕ ПО ТЕМАМ (РАЗДЕЛАМ) С УКАЗАНИЕМ ОТВЕДЕННОГО НА НИХ КОЛИЧЕСТВА АКАДЕМИЧЕСКИХ ЧАСОВ И ВИДОВ УЧЕБНЫХ ЗАНЯТИЙ

4.1. Содержание дисциплины, структурированное по темам (разделам)

Тема 1. Основы баз данных.

Основные понятия БД. Модели БД. Уровни моделирования. Виды инфологических моделей. Виды даталогических моделей. Физические модели. История развития баз данных. Виды баз данных. Картотеки. Сетевые базы данных. Иерархические базы данных. Реляционные базы данных. Многомерные базы данных. Объектно-ориентированные базы данных. Дедуктивные базы данных. NoSQL базы данных. Основные понятия реляционных баз данных. Реляционные системы управления базами данных. Правила Кодда для реляционной СУБД. Отношения, ключи, связи в реляционных

базах данных. Ссылочная целостность данных. Использование языка SQL для создания, модификации и управления данными в реляционной базе данных.

Тема 2. Распределенные базы данных.

Модели архитектуры клиент-сервер. Двухуровневая и многоуровневая архитектуры клиент-сервер. Распределенные базы данных. Примеры архитектур. Методы поддержки распределенных данных. Фрагментация. Репликация. Модели тиражирования. Распределенные ограничения целостности. Распределенные запросы. Распределенные транзакции. Свойства идеальной распределенной БД.

Тема 3. Современные технологии проектирования баз данных.

Автоматизация процесса проектирования БД с использованием CASE-средств. Теория зависимостей. Требования нормализации. Нормализация и нормальные формы. Денормализация. Нисходящая и восходящая денормализация. Денормализация методом слияния таблиц. Внутритабличная денормализация. Денормализация методом «разделяй и властвуй». Оценка сложности проектирования БД. Проектирование БД на инфологическом уровне, даталогическом и физическом уровне. Проблемы проектирования сложных структур баз данных. Проблемы циклических связей в БД. Способы разрешения проблемы. Реализация наследования в БД. Виды наследования. Обычное, взаимоисключающее, законченное, взаимоисключающее законченное наследование в БД. Проблемы обработки данных для рекурсивных связей в однокорневом дереве. Основные нотации для проектирования ER-моделей. Нотации П. Чена, Дж. Мартина, Ч. Баркера, Ж.-Р. Абриаля, IDEF1X. Автоматизация процесса проектирования БД с использованием CASE-средств. Генерация SQL-скрипта для создания базы данных. Прямое и обратное проектирование.

Тема 4. Реализация типовых задач баз данных в современных СУБД.

Основные возможности современных СУБД. Преимущества и недостатки работы в корпоративных СУБД Microsoft SQL Server, Oracle Database, PostgreSQL. Встраиваемые кроссплатформенные СУБД. Основные возможности диалекта Transact-SQL от СУБД Microsoft SQL Server. Процедурное расширение языка PL/SQL в СУБД Oracle Database. Основные возможности диалекта PL/pgSQL в СУБД PostgreSQL. Анализ современных СУБД для решения типовых задач баз данных. Использование XML для хранения данных.. Основные методы для работы с типом данных XML в СУБД Microsoft SQL Server.

4.2. Разделы дисциплины и трудоемкость по видам учебных занятий (в академических часах).

Название раздела	Общая трудоемкость, всего часов	Контактная работа обучающихся с преподавателем						Самостоятельная работа обучающихся		Контроль
		все-го	лекции	практические работы	лабораторные работы	консультации	иные виды контактной работы	КР/КП	самостоятельные занятия	
Тема 1. Основы баз данных	40,3	12	4	4	4			2	13,3	13
Тема 2. Распределенные базы данных	36	12	4	4	4			2	10	12
Тема 3. Современные технологии проектирования баз данных	56	12	4	4	4			4	26	14
Тема 4. Реализация типовых задач баз данных в современных СУБД	45,05	12	4	4	4			3,7	15	14,35
Защита курсовой	0,3	0,3					0,3			

Название раздела	Общая трудоемкость, всего часов	Контактная работа обучающихся с преподавателем						Самостоятельная работа обучающихся		Контроль
		все-го	лек-ции	практические работы	лабораторные работы	консультации	иные виды контактной работы	КР/КП	самостоятельные занятия	
работы										
Консультации к экзамену	2	2				2				
Экзамен	0,35	0,35					0,35			
Итого	180	50,65	16	16	16	2	0,65	11,7	64,3	53,35

Виды лабораторных, практических и самостоятельных работ

Тема	Вид занятий*	Содержания	Часы
Тема 1. Основы баз данных.	ПР	Практические занятия на тему: «Основные понятия реляционных БД», «Выявление ограничений целостности в таблицах БД».	2 2
	ЛР	Лабораторные занятия на тему: «Разработка запросов для создания таблиц на языке SQL»	4
	СР	Изучение конспекта лекций. Изучение теоретического материала по источникам. Подготовка к практическим занятиям. Подготовка к лабораторным работам.	5 5 3,3
Тема 2. Распределенные базы данных.	ПР	Практические занятия на тему: «Основные понятия реляционных БД», «Выявление ограничений целостности в таблицах БД».	2 2
	ЛР	Лабораторные занятия на тему: «Разработка запросов к отдельным и связанным таблицам на языке SQL».	4
	СР	Изучение конспекта лекций. Изучение теоретического материала по источникам. Подготовка к практическим занятиям. Подготовка к лабораторным работам.	2 2 4 2
Тема 3. Современные технологии проектирования баз данных.	ПР	Практические занятия на тему: «Теория нормализации», «Основные нотации для проектирования ER-моделей: П. Чена, Дж. Мартина».	2 2
	ЛР	Лабораторные занятия на тему: «Проектирование БД на инфологическом уровне, даталогическом и физическом уровне».	4
	СР	Изучение конспекта лекций. Изучение теоретического материала по источникам. Подготовка к практическим занятиям. Подготовка к лабораторным работам.	4 10 6 6
Тема 4. Реализация типовых задач баз данных в современных СУБД.	ПР	Практические занятия на тему: «Процедурное расширение языка PL/pgSQL в СУБД PostgreSQL».	4
	ЛР	Лабораторные занятия на тему: «Основные возможности диалекта Transact-SQL от СУБД Microsoft SQL Server».	4
	СР	Изучение конспекта лекций. Изучение теоретического материала по источникам.	4 3

Тема	Вид занятий*	Содержания	Часы
		Подготовка к практическим занятиям.	4
		Подготовка к лабораторным работам.	4
Курсовая работа	СР	Подготовка курсовой работы	11,7

* СР – самостоятельная работа, ЛБ – лабораторные работы, ПЗ – практические занятия

Курсовые работы выполняются каждым обучающимся на одну общую тему «Разработка базы данных для ... (*предметной области*)». В качестве предметной области может быть выбран процесс в органах государственной или муниципальной власти, область социальной сферы или общественной жизни. Требования к содержанию и структуре курсовой работы изложены в Методическом обеспечении дисциплины.

5. ПЕРЕЧЕНЬ УЧЕБНО-МЕТОДИЧЕСКОГО ОБЕСПЕЧЕНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ

1. Громов А.Ю. Современные технологии разработки интегрированных информационных систем: учеб. пособие / А.Ю. Громов, Н.Н. Гринченко, Н.В. Шемонаев; Рязан. гос. радиотехн. ун-т. - Рязань, 2015. - 48 с. <https://elibrsreu.ru/ebs/download/562>

2. Гринченко Н.Н. Инструментальные средства поддержки проектирования баз данных: учеб. пособие / Н.Н. Гринченко, А.Ю. Громов; Рязан. гос. радиотехн. ун-т. - Рязань, 2015. - 48 с. <https://elibrsreu.ru/ebs/download/731>

6. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)

Оценочные материалы приведены в Приложении к рабочей программе дисциплины (см. документ «Оценочные материалы по дисциплине «Современные технологии баз данных»).

7. ПЕРЕЧЕНЬ ОСНОВНОЙ И ДОПОЛНИТЕЛЬНОЙ УЧЕБНОЙ ЛИТЕРАТУРЫ, НЕОБХОДИМОЙ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ (МОДУЛЯ)

а) основная учебная литература:

1. Громов А.Ю. Современные технологии разработки интегрированных информационных систем: учеб. пособие / А.Ю. Громов, Н.Н. Гринченко, Н.В. Шемонаев; Рязан. гос. радиотехн. ун-т. - Рязань, 2015. - 48 с. <https://elibrsreu.ru/ebs/download/562>

2. Клиент-серверные приложения баз данных: учеб. пособие / А.В. Благодаров, Н.Н. Гринченко, А.Ю. Громов; Рязан. гос. радиотехн. ун-т. Рязань, 2017. 72 с. <https://elibrsreu.ru/ebs/download/2356>

3. Гринченко Н.Н. Инструментальные средства поддержки проектирования баз данных: учеб. пособие / Н.Н. Гринченко, А.Ю. Громов; Рязан. гос. радиотехн. ун-т. - Рязань, 2015. - 48 с. <https://elibrsreu.ru/ebs/download/731>

4. Пржиялковский В.В. Введение в Oracle SQL [Электронный ресурс]/ Пржиялковский В.В.- Электрон. текстовые данные. - М.: Интернет-Университет Информационных Технологий (ИНТУИТ), 2011. - 219 с. - URL: <http://www.iprbookshop.ru/16692>

5. Липаев В.В. Программная инженерия сложных заказных программных продуктов [Электронный ресурс]: учебное пособие/ Липаев В.В.- Электрон. текстовые данные. - М.: МАКС Пресс, 2014. - 309 с. - URL: <http://www.iprbookshop.ru/27297>

6. Введение в программные системы и их разработку [Электронный ресурс]/ С.В. Назаров [и др.]. - Электрон. текстовые данные.- М.: Интернет-Университет Информационных Технологий (ИНТУИТ), 2016. - 649 с. - URL: <http://www.iprbookshop.ru/52145>

7. Волкова Т.В. Разработка систем распределенной обработки данных [Электронный ресурс]: учебно-методическое пособие/ Волкова Т.В., Насейкина Л.Ф.- Электрон. текстовые данные. - Оренбург: Оренбургский государственный университет, ЭБС АСВ, 2012. - 330 с. - URL: <http://www.iprbookshop.ru/30127>

8. Сергеев С.В. Разработка и проектирование Web-приложений в Oracle Developer

[Электронный ресурс] : учебное пособие / С.В. Сергеенко. — Электрон. текстовые данные. — Москва, Саратов: Интернет-Университет Информационных Технологий (ИНТУИТ), Вузовское образование, 2017. — 456 с. — 978-5-4487-0091-0. — Режим доступа: <http://www.iprbookshop.ru/67374.html>

б) дополнительная учебная литература:

9. Благодаров А.В., Гринченко Н.Н., Громов А.Ю. Базы данных. Разработка клиентских приложений на платформе .NET. Рязань, 2017. 231 с. <https://elib.rsreu.ru/ebs/download/2355>

10. Снетков В.М. Практикум прикладного программирования на С# в среде VS.NET 2008 [Электронный ресурс]/ Снетков В.М. - Электрон. текстовые данные. - М.: Интернет-Университет Информационных Технологий (ИНТУИТ), 2010. - 608 с. - Режим доступа: <http://www.iprbookshop.ru/16728>

11. Королева О.Н. Базы данных [Электронный ресурс]: курс лекций/ Королева О.Н., Мажукин А.В., Королева Т.В.- Электрон. текстовые данные. - М.: Московский гуманитарный университет, 2012. - 66 с. - URL: <http://www.iprbookshop.ru/14515>

12. Суркова Н.Е. Проектирование информационных систем [Электронный ресурс]: методические указания к курсовому проекту/ Суркова Н.Е.- Электрон. текстовые данные. - М.: Российский новый университет, 2010. - 60 с. - URL: <http://www.iprbookshop.ru/21303>

13. Стешин А.И. Информационные системы в организации [Электронный ресурс]: учебное пособие/ Стешин А.И. - Электрон. текстовые данные. - Саратов: Вузовское образование, 2013. - 194 с. - URL: <http://www.iprbookshop.ru/16346>

8. ПЕРЕЧЕНЬ РЕСУРСОВ ИНФОРМАЦИОННО–ТЕЛЕКОММУНИКАЦИОННОЙ СЕТИ «ИНТЕРНЕТ», НЕОБХОДИМЫХ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Обучающимся предоставлена возможность индивидуального доступа к следующим электронно-библиотечным системам:

– Электронно-библиотечная система «IPRbooks», режим доступа – с любого компьютера РГРТУ без пароля, из сети интернет по паролю. – URL: <https://iprbookshop.ru/>.

– Электронная библиотека РГРТУ, режим доступа – с любого компьютера РГРТУ без пароля, из сети интернет по паролю. – URL: <https://elib.rsreu.ru/>

9. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

Указания в рамках лекций

Во время лекции студент должен вести краткий конспект.

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.

Обучающимся необходимо регулярно отводить время для повторения пройденного материала, проверяя свои знания, умения и навыки по контрольным вопросам.

Указания в рамках практических (семинарских) занятий

Практические (семинарские) занятия составляют важную часть профессиональной подготовки студентов. Основная цель проведения практических (семинарских) занятий – формирование у студентов аналитического и творческого мышления путем приобретения практических навыков.

Методические указания к практическим (семинарским) занятиям по дисциплине наряду с рабочей программой и графиком учебного процесса относятся к методическим документам, определяющим уровень организации и качества образовательного процесса. Содержание практических занятий фиксируется в рабочей программе дисциплины в разделе 4.

Важнейшей составляющей любой формы практических занятий являются упражнения (задания). Основа в упражнении – пример, который разбирается с позиций теории, развитой в лекции.

Как правило, основное внимание уделяется формированию конкретных умений, навыков, что и определяет содержание деятельности студентов – решение задач, графические работы, уточнение категорий и понятий науки, являющихся предпосылкой правильного мышления и речи.

Практические (семинарские) занятия выполняют следующие задачи:

- стимулируют регулярное изучение рекомендуемой литературы, а также внимательное отношение к лекционному курсу;
- закрепляют знания, полученные в процессе лекционного обучения и самостоятельной работы над литературой;
- расширяют объем профессионально значимых знаний, умений и навыков;
- позволяют проверить правильность ранее полученных знаний;
- прививают навыки самостоятельного мышления, устного выступления;
- способствуют свободному оперированию терминологией;
- представляют преподавателю возможность систематически контролировать уровень самостоятельной работы студентов.

При подготовке к практическим (семинарским) занятиям необходимо просмотреть конспекты лекций и методические указания, рекомендованную литературу по данной теме, а так же подготовиться к ответу на контрольные вопросы.

В ходе выполнения индивидуального задания практического занятия студент готовит отчет о работе (с помощью офисного пакета Open Office или другом редакторе доступном студенту). В отчет заносятся результаты выполнения каждого пункта задания (анализ задачи, найденные пути решения, поясняющие схемы, диаграммы, графики, таблицы, расчеты, ответы на вопросы пунктов задания, выводы по проделанной работе и т.д.). Примерный образец оформления отчета предоставляется студентам в виде раздаточных материалов или прилагается к рабочей программе дисциплины.

За 10 минут до окончания занятия преподаватель проверяет объем выполненной за занятие работы и отмечает результат в рабочем журнале. Оставшиеся невыполненными пункты задания практического занятия студент обязан доделать самостоятельно.

После проверки отчета преподаватель может проводить устный или письменный опрос студентов для контроля усвоения ими основных теоретических и практических знаний по теме занятия (студенты должны знать смысл полученных ими результатов и ответы на контрольные вопросы). По результатам проверки отчета и опроса выставляется оценка за практическое занятие.

Указания в рамках лабораторных работ

Лабораторные работы 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.

Состав заданий для лабораторной работы должен быть спланирован с таким расчетом, чтобы за отведенное время они могли быть качественно выполнены большинством студентов.

Необходимыми структурными элементами лабораторной работы, помимо самостоятельной деятельности студентов, являются инструктаж, проводимый преподавателем, а так же организация обсуждения итогов выполнения лабораторной работы.

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

Порядок проведения лабораторных работ в целом совпадает с порядком проведения практических занятий. Помимо выполнения работы для каждой лабораторной работы предусмотрена процедура защиты, в ходе которой преподаватель проводит устный или письменный опрос студентов для контроля понимания выполненных ими измерений, правильной интерпретации полученных результатов и усвоения ими основных теоретических и практических знаний по теме занятия.

Указания в рамках подготовки к промежуточной аттестации

При подготовке к экзамену в дополнение к изучению конспектов лекций, учебных пособий, слайдов и другого раздаточного материала, предусмотренного рабочей программой дисциплины, необходимо пользоваться учебной литературой, рекомендованной к настоящей рабочей программе. При подготовке к экзамену нужно изучить теорию: определения всех понятий и подходы к

оцениванию до состояния понимания материала и самостоятельно решить по нескольким типовым задачам из каждой темы (в том случае, если тема предусматривает решение задач). При решении задач всегда необходимо уметь качественно интерпретировать итог решения.

Указания в рамках самостоятельной работы студентов

Самостоятельная работа студентов по дисциплине играет важную роль в ходе всего учебного процесса. Самостоятельное изучение тем учебной дисциплины способствует:

- закреплению знаний, умений и навыков, полученных в ходе аудиторных занятий;
- углублению и расширению знаний по отдельным вопросам и темам дисциплины;
- освоению умений прикладного и практического использования полученных знаний.

Самостоятельная работа как вид учебной работы может использоваться на лекциях, лабораторных и практических занятиях, а также имеет самостоятельное значение – внеаудиторная самостоятельная работа обучающихся – при подготовке к лекциям, лабораторным и практическим занятиям, а также к экзамену.

Основными видами самостоятельной работы по дисциплине являются:

- самостоятельное изучение отдельных вопросов и тем дисциплины;
- выполнение лабораторного задания;
- подготовка к защите лабораторного задания, оформление отчета;
- выполнение практического задания;
- подготовка к защите практического задания, оформление отчета;
- оформление пояснительной записки к курсовой работе/проекту.

Рекомендации по работе с литературой

Теоретический материал курса становится более понятным, когда дополнительно к прослушиванию лекции и изучению конспекта, изучается дополнительная рекомендованная литература. Литературу по курсу рекомендуется изучать в библиотеке, с использованием доступной электронной библиотечной системы или с помощью сети Интернет (источники, которые могут быть использованы без нарушения авторских прав).

10. ПЕРЕЧЕНЬ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, ИСПОЛЬЗУЕМЫХ ПРИ ОСУЩЕСТВЛЕНИИ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ (МОДУЛЮ), ВКЛЮЧАЯ ПЕРЕЧЕНЬ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ И ИНФОРМАЦИОННО-СПРАВОЧНЫХ СИСТЕМ

При проведении занятий по дисциплине используются следующие информационные технологии:

- удаленные информационные коммуникации между студентами и преподавателем, ведущим лекционные, лабораторные и практические занятия, посредством информационной образовательной среды ФГБОУ ВО «РГРТУ», позволяющие осуществлять оперативный контроль графика выполнения и содержания образовательного процесса, решение организационных вопросов, консультирование;
- поиск актуальной научной, статистической и общественно-политической информации для выполнения самостоятельной работы и контрольных заданий;
- доступ к современным профессиональным базам данных (в том числе международным реферативным базам данных научных изданий) и информационным справочным системам;
- проведение аудиторных занятий с использованием презентаций и раздаточных материалов в электронном виде;
- выполнение студентами различных видов учебных работ с использованием лицензионного программного обеспечения, установленного на рабочих местах студента в компьютерных классах и в помещениях для самостоятельной работы, а также для выполнения самостоятельной работы в домашних условиях.

Перечень лицензионного программного обеспечения:

- 1) операционная система Windows;

- 2) Kaspersky Endpoint Security;
- 3) LibreOffice, лицензия LGPLv3.
- 4) Microsoft SQL Server не ниже 2008 (лицензия Microsoft DreamSpark Membership ID 700102019);
- 5) Microsoft Office Visio (лицензия Microsoft DreamSpark Membership ID 700102019);
- 6) PostgreSQL Database Management System (лицензия PostgreSQL License).

Перечень профессиональных баз данных (в том числе международным реферативным базам данных научных изданий) и информационных справочных систем:

- Информационно-правовой портал ГАРАНТ.РУ [Электронный ресурс]. – URL: <http://www.garant.ru>. – Режим доступа: свободный доступ.
- Справочная правовая система КонсультантПлюс [Электронный ресурс]. – URL: <http://www.consultant.ru/online/>. – Режим доступа: свободный доступ (будние дни – 20.00-24.00, выходные и праздничные дни – круглосуточно);

11. ОПИСАНИЕ МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЙ БАЗЫ, НЕОБХОДИМОЙ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ (МОДУЛЮ)

Наименование специальных помещений и помещений для самостоятельной работы	Оснащенность специальных помещений и помещений для самостоятельной работы	Перечень лицензированного программного обеспечения. Реквизиты подтверждающего документа
<p>Учебно-административный корпус, а. 122 Учебная аудитория для проведения занятий лекционного и семинарского типа, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, лабораторных работ, практических и самостоятельных занятий</p>	<p>56 посадочных места, компьютерная техника (10ПК - Компьютер PC-120 16 ЕДО, Персональный компьютер на базе intel Celeron D, ПЭВМ "Celeron-1700", ПЭВМ "Pentium", ПЭВМ C2,0/256/40/CD-RW/FDD/17"/клав.+мышь отп./pad – 4шт, ПЭВМ Celeron-366, ПЭВМ P-120) с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду РГРТУ, мультимедийное оборудование (проектор NEC, экран), интерактивная доска, специализированная мебель (стулья-56, столы-23), доска</p>	<p>Microsoft Windows (Microsoft Imagine, номер подписки 700102019, бессрочно) Microsoft Visual Studio (Microsoft Imagine, номер подписки 700102019, бессрочно) Microsoft Visio (Microsoft Imagine, номер подписки 700102019, бессрочно) Microsoft SQL Server (Microsoft Imagine, номер подписки 700102019, бессрочно) Microsoft Project (Microsoft Imagine, номер подписки 700102019, бессрочно) Microsoft Access (Microsoft Imagine, номер подписки 700102019, бессрочно) Microsoft Virtual PC (Microsoft Imagine, номер подписки 700102019, бессрочно) OpenSUSE GNU GPL v2 Свободно распространяемое программное обеспечение Mozilla Firefox (Mozilla Public License (MPL)) LibreOffice (LGPLv3) Business Studio 4.0 (Business Studio (ДемOVERсия)) Adobe Reader (Adobe Personal Computer Software License Agreement) GIMP 2 (GIMP Software EULA. GNU GPL) Sun VirtualBox (GNU GPL VirtualBox Personal Use and Evaluation License (PUEL)); VMware Player (VMWARE EULA); 7-Zip File Manager (GNU LGPL + BSD 3-clause License)</p>
<p>Учебно-административный корпус, а. 209 Помещение для</p>	<p>48 посадочных мест, компьютерная техника (14ПК - ПЭВМ</p>	<p>Microsoft Windows (Microsoft Imagine, номер подписки 700102019, бессрочно) Microsoft Visual Studio (Microsoft</p>

Наименование специальных помещений и помещений для самостоятельной работы	Оснащенность специальных помещений и помещений для самостоятельной работы	Перечень лицензированного программного обеспечения. Реквизиты подтверждающего документа
<p>самостоятельной работы</p>	<p>C2,0/256/40/CD-RW/FDD/17"/клав.+мышь отп./pad, ПК HP 8200 (Системный блок) - 14 шт., Монитор Samsung S22A200B - 5 шт., Монитор Samsung EX2220 - 9шт.) с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду РГРТУ, мультимедийное оборудование (проектор Ben Q, экран), специализированная мебель (стулья-48, столы-24), доска</p>	<p>Imagine, номер подписки 700102019, бессрочно) Microsoft Visio (Microsoft Imagine, номер подписки 700102019, бессрочно) Microsoft SQL Server (Microsoft Imagine, номер подписки 700102019, бессрочно) Microsoft Project (Microsoft Imagine, номер подписки 700102019, бессрочно) Microsoft Access (Microsoft Imagine, номер подписки 700102019, бессрочно) Microsoft Virtual PC (Microsoft Imagine, номер подписки 700102019, бессрочно) OpenSUSE GNU GPL v2 Свободно распространяемое программное обеспечение(Mozilla Firefox (Mozilla Public License (MPL)) LibreOffice (LGPLv3) Business Studio 4.0 (Business Studio (Демоверсия)) Sun VirtualBox (GNU GPL VirtualBox Personal Use and Evaluation License (PUEL)) Far Manager 3 (Far Manager BSD License) Notepad++ (GPL) Tomcat (Apache License Version 2.0)</p>

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

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

Кафедра «Государственного, муниципального и корпоративного управления»

ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ПО ДИСЦИПЛИНЕ

Б1.В.ДВ.02.01 «Современные технологии баз данных»

Направление подготовки
38.04.04 «Государственное и муниципальное управление»

Профиль – Информационные технологии в государственном и муниципальном
управлении

ОПОП академической магистратуры
«Государственное и муниципальное управление»

Формы обучения – очно-заочная

1. ОБЩИЕ ПОЛОЖЕНИЯ

Оценочные материалы – это совокупность учебно-методических материалов (практических заданий, описаний форм и процедур проверки), предназначенных для оценки качества освоения обучающимися данной дисциплины как части ОПОП.

Цель – оценить соответствие знаний, умений и владений, приобретенных обучающимся в процессе изучения дисциплины, целям и требованиям ОПОП в ходе проведения промежуточной аттестации.

Основная задача – обеспечить оценку уровня сформированности компетенций.

Контроль знаний обучающихся проводится в форме промежуточной аттестации.

Промежуточная аттестация проводится в форме экзамена, защиты курсовой работы. Форма проведения экзамена - тестирование, письменный опрос по теоретическим вопросам и выполнение практических заданий.

2. ПАСПОРТ ОЦЕНОЧНЫХ МАТЕРИАЛОВ ПО ДИСЦИПЛИНЕ

<i>Контролируемые разделы (темы) дисциплины</i>	<i>Код контролируемой компетенции (или её части)</i>	<i>Вид, метод, форма оценочного мероприятия</i>
Тема 1. Основы баз данных	ПК-11	Экзамен Курсовая работа
Тема 2. Распределенные базы данных	ПК-11	Экзамен Курсовая работа
Тема 3. Современные технологии проектирования баз данных	ПК-11	Экзамен Курсовая работа
Тема 4. Реализация типовых задач баз данных в современных СУБД	ПК-11	Экзамен Курсовая работа

3. ОПИСАНИЕ ПОКАЗАТЕЛЕЙ И КРИТЕРИЕВ ОЦЕНИВАНИЯ КОМПЕТЕНЦИЙ

Сформированность каждой компетенции (или ее части) в рамках освоения данной дисциплины оценивается по трехуровневой шкале:

1) пороговый уровень является обязательным для всех обучающихся по завершении освоения дисциплины;

2) продвинутый уровень характеризуется превышением минимальных характеристик сформированности компетенций по завершении освоения дисциплины;

3) эталонный уровень характеризуется максимально возможной выраженностью компетенций и является важным качественным ориентиром для самосовершенствования.

Уровень освоения компетенций, формируемых дисциплиной:

а) описание критериев и шкалы оценивания тестирования:

Шкала оценивания	Критерий
3 балла (эталонный уровень)	уровень усвоения материала, предусмотренного программой: процент верных ответов на тестовые вопросы от 85 до 100%
2 балла (продвинутый уровень)	уровень усвоения материала, предусмотренного программой: процент верных ответов на тестовые вопросы от 70 до 84%
1 балл (пороговый уровень)	уровень усвоения материала, предусмотренного программой: процент верных ответов на тестовые вопросы от 50 до 69%
0 баллов	уровень усвоения материала, предусмотренного программой: процент верных ответов на тестовые вопросы от 0 до 49%

б) описание критериев и шкалы оценивания теоретического вопроса:

Шкала оценивания	Критерий
3 балла	выставляется студенту, который дал полный ответ на вопрос,

Шкала оценивания	Критерий
(эталонный уровень)	показал глубокие систематизированные знания, смог привести примеры, ответил на дополнительные вопросы преподавателя
2 балла (продвинутый уровень)	выставляется студенту, который дал полный ответ на вопрос, но на некоторые дополнительные вопросы преподавателя ответил только с помощью наводящих вопросов
1 балл (пороговый уровень)	выставляется студенту, который дал неполный ответ на вопрос в билете и смог ответить на дополнительные вопросы только с помощью преподавателя
0 баллов	выставляется студенту, который не смог ответить на вопрос

в) описание критериев и шкалы оценивания практического задания:

Шкала оценивания	Критерий
3 балла (эталонный уровень)	Задание решено верно
2 балла (продвинутый уровень)	Задание решено верно, но имеются неточности в логике решения
1 балл (пороговый уровень)	Задание решено верно, с дополнительными наводящими вопросами преподавателя
0 баллов	Задание не решено

г) описание критериев и шкалы оценивания курсовой работы:

Шкала оценивания	Критерий
Оценка «отлично» (эталонный уровень)	курсовая работа выполнена в полном объеме, все аналитические этапы и модели выполнены без ошибок, дана оценка полученных результатов, работа выполнено самостоятельно, работа оформлена аккуратно, соблюдались сроки сдачи и защиты курсового проекта, при защите курсового проекта студент ответил на все предложенные вопросы
Оценка «хорошо» (продвинутый уровень)	курсовая работа выполнена в полном объеме, присутствуют незначительные ошибки при проведении анализа и/или при построении моделей, дана оценка полученных результатов, работа выполнена самостоятельно, работа оформлена аккуратно, соблюдались сроки сдачи и защиты курсового проекта, при защите курсового проекта студент ответил не на все предложенные вопросы (правильных ответов не менее 75%)
Оценка «удовлетворительно» (пороговый уровень)	курсовая работа выполнена в полном объеме, присутствуют ошибки при проведении анализа и/или при построении моделей, оценка полученных результатов не является полной, работа выполнена самостоятельно, по оформлению работы имеются замечания, частично соблюдались сроки сдачи и защиты курсового проекта, при защите курсового проекта студент ответил не на все предложенные вопросы (правильных ответов не менее 50%)
Оценка «неудовлетворительно»	курсовая работа выполнена не в полном объеме, присутствуют грубые ошибки при проведении анализа и/или при построении моделей, отсутствует оценка полученных результатов, работа выполнена не самостоятельно, по оформлению работы имеются замечания, не соблюдались сроки сдачи и защиты курсового проекта, при защите курсового проекта студент ответил не на все предложенные вопросы (правильных ответов менее 50%)

На промежуточную аттестацию в форме экзамена выносятся тест, два теоретических вопроса и 2 задачи. Максимально студент может набрать 15 баллов. Итоговый суммарный балл студента, полученный при прохождении промежуточной аттестации, переводится в традиционную

форму по системе «отлично», «хорошо», «удовлетворительно» и «неудовлетворительно».

Оценка «отлично» выставляется студенту, который набрал в сумме 15 баллов (выполнил все задания на эталонном уровне). Обязательным условием является выполнение всех предусмотренных в течение семестра практических заданий и лабораторных работ.

Оценка «хорошо» выставляется студенту, который набрал в сумме от 10 до 14 баллов при условии выполнения всех заданий на уровне не ниже продвинутого. Обязательным условием является выполнение всех предусмотренных в течение семестра практических заданий и лабораторных работ.

Оценка «удовлетворительно» выставляется студенту, который набрал в сумме от 5 до 9 баллов при условии выполнения всех заданий на уровне не ниже порогового. Обязательным условием является выполнение всех предусмотренных в течение семестра практических заданий и лабораторных работ.

Оценка «неудовлетворительно» выставляется студенту, который набрал в сумме менее 5 баллов или не выполнил всех предусмотренных в течение семестра практических заданий и лабораторных работ.

4 ТИПОВЫЕ КОНТРОЛЬНЫЕ ЗАДАНИЯ ИЛИ ИНЫЕ МАТЕРИАЛЫ

4.1. Промежуточная аттестация в форме экзамена

<i>Коды компетенций</i>	<i>Содержание компетенций</i>
ПК-11	способность осуществлять верификацию и структуризацию информации, получаемой из разных источников

а) типовые тестовые вопросы:

1. Проектирование баз данных – это:
 - а) процесс создания проекта базы данных, предназначенный для поддержки функционирования предприятия и способствующий достижению его целей
 - б) выбор СУБД подходящего типа, предназначенной для поддержки создаваемого приложения базы данных
 - в) проектирование интерфейса пользователя и прикладных программ, предназначенных для работы с базой данных
 - г) подготовительные действия, позволяющие с максимально возможной эффективностью реализовать этапы жизненного цикла приложений баз данных
2. На каком этапе производят оценку показателей существующих информационных систем с целью выявления их сильных и слабых сторон?
 - а) Проектирование базы данных
 - б) Администрирование баз данных
 - в) Тестирование
 - г) Планирование разработки базы данных
3. В каком порядке следуют уровни проектирования БД?
 - а) физический, логический, концептуальный
 - + б) концептуальный, логический, физический
 - в) внешний, физический, концептуальный
 - г) концептуальный, физический, логический
4. Какой элемент не используется в рамках модели «entity-relationship»?
 - а) узел
 - б) атрибут
 - в) связь
 - г) сущность
5. ER-диаграмма - это
 - + а) графическая модель предметной области
 - б) диаграмма распределения ресурсов СУБД
 - в) обязательный этап проектирования БД

- г) средство установления связей между таблицами БД
6. Какие виды связей существуют в рамках модели «entity-relationship»?
- + а) «многие-ко-многим», «один-к-одному», «один-ко-многим»
 - б) «один-к-одному», «один-ко-многим»
 - в) «многие-ко-многим», «один-к-одному»,
 - г) «многие-ко-многим», «один-ко-многим»
7. Модель «сущность-связь» была предложена:
- а) К. Дейтом
 - б) Э. Коддом
 - + в) П. Ченом
 - г) Р. Бойсом
8. Дополните утверждение: «Переменная отношения находится в _____ тогда и только тогда, когда каждая её нетривиальная и неприводимая слева функциональная зависимость имеет в качестве своего детерминанта некоторый потенциальный ключ».
- а) первой нормальной форме
 - + б) нормальной форме Бойса-Кодда
 - в) доменно-ключевой нормальной форме
 - г) третьей нормальной форме
9. Информационная модель наиболее высокого уровня абстракции разрабатывается на этапе:
- + а) инфологического проектирования
 - б) даталогического проектирования
 - в) физического проектирования
 - г) на всех этапах модель имеет одинаковый уровень абстракции
10. Схема базы данных для конкретной СУБД разрабатывается на этапе
- а) инфологического проектирования
 - б) даталогического проектирования
 - + в) физического проектирования
 - г) на любом из этих этапов
11. Какая из этих технологий доступа к данным не принадлежит Microsoft?
- а) ADO
 - б) ODBC
 - + в) BDE
 - г) OLE DB
12. Под внешней схемой принято понимать
- а) логическую конструкцию информационной модели предметной области
 - б) внутреннюю схему базы данных, трансформированную под представления пользователя о базе данных
 - + в) такую организацию представления данных в базе данных, которое наиболее естественным и простым способом отражало бы взгляд пользователей на эти данные, когда они их обрабатывают
 - г) альтернативную к внутренней схеме модель физической организации базы данных
13. Какие объекты не являются входными данными для проектирования базы данных
- + а) спецификации модулей приложений базы данных
 - б) информационная модель предметной области базы данных
 - в) бизнес-правила
 - г) функциональная модель предметной области базы данных
14. Объект, соединяющий базу данных с объектом DataSet в ADO.NET – это
- а) Connection
 - + б) DataAdapter
 - в) DataReader
 - г) TableAdapter
15. В рамках физического проектирования РБД вертикальное разбиение таблицы — это
- + а) процесс перемещения некоторых колонок таблицы в другую новую таблицу, которая имеет тот же первичный ключ, что и исходная таблица
 - б) процесс декомпозиции таблицы на две или более таблиц с целью устранения частичной зависимости неключевых колонок от составного первичного ключа
 - в) процесс создания двух независимых таблиц из одной таблицы

- г) процесс создания независимых таблиц посредством намеренного дублирования колонок одной таблицы в другой
16. Нормализация отношений информационной модели предметной области является
- а) распределением атрибутов по отношениям базы данных
 - б) группировкой атрибутов в отношении предметной области
 - + в) механизмом создания логической модели реляционной базы данных
 - г) приведением отношений к БКНФ
17. На ER-диаграмме отношение отображается при помощи
- + а) прямой линии
 - б) ромба
 - в) прямоугольника
 - г) пунктирной линии
18. СА ERWin предназначено для разработки с выполнением верификации по нотации:
- а) IDEF0
 - б) DFD
 - в) EXPRESS
 - г) IDEF1X и IE
19. Какой группы операторов не существует в языке SQL?
- а) DML
 - + б) DPL
 - в) DDL
 - г) DCL
20. Для создания таблиц в языке SQL используются операторы из группы:
- а) DML
 - + б) DDL
 - в) DCL
 - г) TCL
21. Операторы CREATE TABLE и DROP TABLE относятся к группе операторов
- а) Transaction Control
 - б) DML
 - + в) DDL
 - г) DCL
22. Операторы INSERT, DELETE, UPDATE относятся к группе операторов
- + а) DML
 - б) DCL
 - в) DDL
 - г) Transaction Control
23. Операторы COMMIT и ROLLBACK относятся к группе операторов
- а) DML
 - б) DCL
 - в) DDL
 - + г) Transaction Control
24. Какого вида БД не существует
- а) Картоотеки
 - + б) Библиотеки
 - в) Сетевые базы данных
 - г) Многомерные базы данных
25. Какие уровни моделирования БД не существуют
- а) инфологический
 - б) даталогический
 - в) процедурный
 - + г) физический
26. Что означает термин NoSQL БД
- + а) не только реляционные БД
 - б) только не реляционные БД
 - в) постреляционные
 - г) термин не имеет отношение к БД

27.СУБД должна удовлетворять 12 правилам:

- + а) Кодда
- б) Бойса-Кодда
- в) Бэкуса-Наэра
- г) Паскаля

28.Правила Кодда гласят, что _____ должна удовлетворять 12 правилам:

- + а) СУБД
- б) клиентское приложение
- в) БД
- г) операционная система

29.Какого правила Кодда не существуют

- + а) историческая независимость данных
- б) физическая независимость данных
- в) логическая независимость данных
- г) независимость контроля целостности

б) типовые практические задания:

Задание 1

Разработать сценарий на языке SQL, который включал бы все операторы подмножества языка DDL. Задание должно быть реализовано с помощью любой доступной обучающемуся СУБД.

Задание 2

Разработать сценарий на языке SQL, который включал бы все операторы подмножества языка DML. Задание должно быть реализовано с помощью любой доступной обучающемуся СУБД.

Задание 3

Разработать сценарий на языке SQL, который включал бы все операторы подмножества языка DCL. Задание должно быть реализовано с помощью любой доступной обучающемуся СУБД.

Задание 4

Разработать сценарий на языке SQL, который включал бы все операторы подмножества языка TCL. Задание должно быть реализовано с помощью любой доступной обучающемуся СУБД.

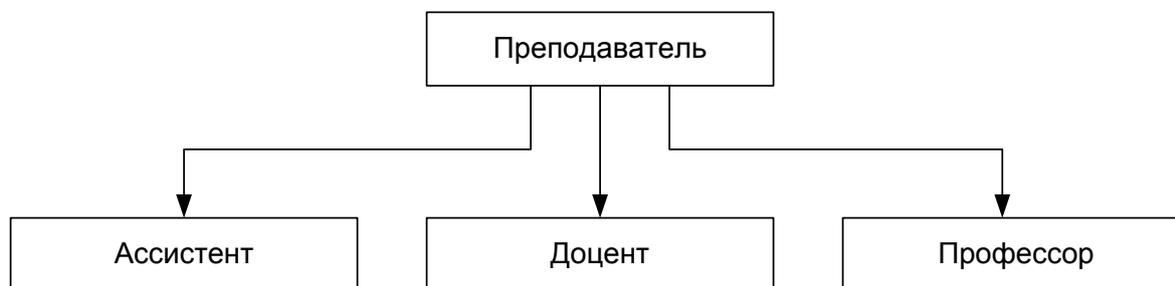
Критерии выполнения задания 1-4

Задание считается выполненным, если обучающийся разработал работоспособный сценарий на одном из диалектов языка SQL, который включает все операторы требуемого подмножества языка SQL.

Задание 5

Пусть существует следующая иерархия типов.

Например, преподаватель может быть ассистентом, доцентом.

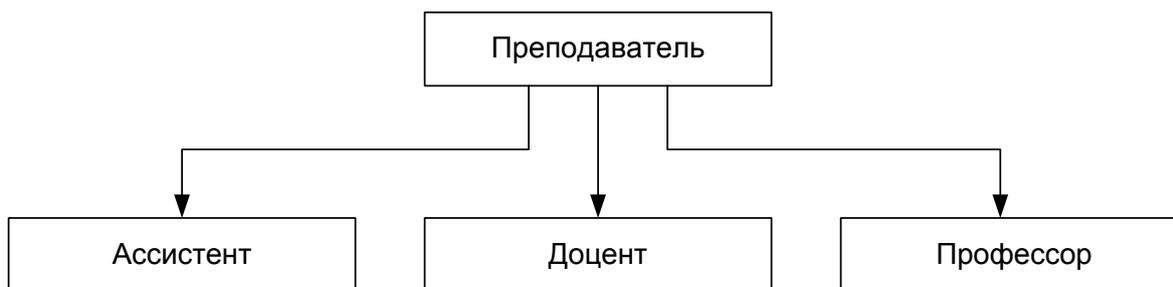


Разработать триггер на языке SQL на добавление и обновление записей для реализации взаимоисключающего наследования.

Задание 6

Пусть существует следующая иерархия типов.

Например, преподаватель может быть ассистентом, доцентом.

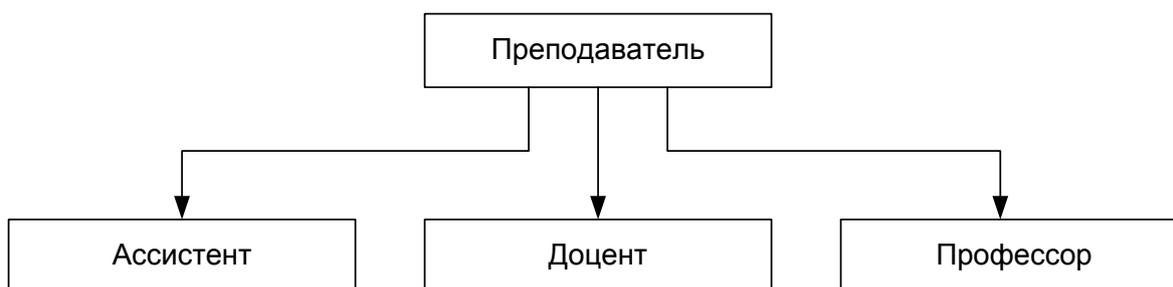


Разработать триггер на языке SQL на и удаление записей для реализации законченного наследования.

Задание 7

Пусть существует следующая иерархия типов.

Например, преподаватель может быть ассистентом, доцентом.



Разработать триггер на языке SQL на добавление и обновление записей для реализации законченного наследования.

Критерии выполнения задания 5-7.

Задание считается выполненным, если обучающийся: разработал работоспособный сценарий на языке SQL, выполняющий заданные действия.

Задание 8.

В соответствии с вариантом задания провести проектирование базы данных для указанной предметной области. Разработать тесты, проверяющие предварительные отношения на соответствие заданной нормальной формы. Разработать сценарий создания базы данных на заданном языке программирования БД.

Варианты для задания 1

№ варианта	Нотация для проектирования ER-модели	Нормальная форма	Диалект языка SQL
1	П. Чена,	3НФ	Transact-SQL
2	Дж. Мартина,	БКНФ	PL/pgSQL
3	Ч. Баркера,	4НФ	PL/SQL
4	Ж.-Р. Абриаля,	3НФ	Transact-SQL
5	IDEF1X	БКНФ	PL/pgSQL
6	П. Чена,	4НФ	PL/SQL
7	Дж. Мартина,	3НФ	Transact-SQL
8	Ч. Баркера,	БКНФ	PL/pgSQL
9	Ж.-Р. Абриаля,	4НФ	PL/SQL
10С	IDEF1X	3НФ	Transact-SQL

Критерии выполнения задания 8.

Задание считается выполненным, если обучающийся: правильно выделил сущности и связи между ними; построил инфологическую модель предметной области, включающую все выделенные сущности и связи, используя для этого указанную в задании нотацию; сформировал набор предварительных отношений в соответствии с правилами перехода; проверил отношения на

соответствие заданной НФ; построил даталогическую модель, разработал сценарий создания базы данных на заданном языке программирования БД.

Варианты предметных областей для практического задания 8

1. База данных хроники восхождений в альпинистском клубе.

Группы альпинистов совершают восхождение на горы разных стран и районов. В базе данных должны записываться даты начала и завершения каждого восхождения, имена и адреса участвовавших в нем альпинистов, название и высота горы, страна и район, где эта гора расположена. Изменение информации о вершине возможно только в случае если на нее не было восхождения. Альпинисты берут с собой на каждое восхождение определенный набор вещей: оборудование разного типа, а так же наборы еды, воду, палатки, обмундирование.

2. База данных медицинского кооператива.

Базу данных использует для работы коллектив врачей. Необходимо хранить имя, пол, дата рождения и домашний адрес каждого их пациента. Всякий раз, когда врач осматривает больного, явившегося к нему на прием, или сам приходит к нему на дом, он записывает дату и место, где проводится осмотр, симптомы, диагноз и предписания больному, проставляет имя пациента, а также свое имя. Если врач прописывает больному какое-либо лекарство, он фиксирует название лекарства, способ его приема, словесное описание предполагаемого действия и возможных побочных эффектов. Также после назначенных предписаний, больной может явиться на сдачу каких-либо анализов, прохождение лечебных процедур или уколов.

3. База данных рыболовной фирмы.

Фирме принадлежит небольшая флотилия рыболовных катеров. Каждый катер имеет паспорт, куда занесены его название, тип и дата постройки. Фирма регистрирует каждый выход на лов, записывая название катера, имена и адреса членов команды с указанием их должностей (капитан, боцман и т.д.), даты выхода и возвращения, а также вес пойманной рыбы отдельно по сортам (например, трески). За время одного рейса катер может посетить несколько банок. Фиксируется дата прихода на каждую банку и дата отплытия, качество выловленной рыбы (отличное, хорошее, плохое). На борту улов взвешивается. Членам команды, которые в течение месяца регулярно привозили улов выше среднего доплачивают надбавку в размере суммы 35% оклада должности каждого члена конкретной команды, деленное на количество членов этой команды.

4. База данных фирмы, проводящей аукционы.

Фирма занимается продажей с аукциона антикварных изделий и произведений искусства. Владельцы вещей, выставляемых на проводимых фирмой аукционах, юридически являются продавцами. Лица, приобретающие эти вещи, именуется покупателями. Получив от продавцов партию предметов, фирма решает, на котором из аукционов выгоднее представить конкретный предмет. Перед проведением очередного аукциона каждой из выставляемых на нем вещей присваивается отдельный номер лота, играющий ту же роль, что и введенный ранее шифр товара. Две вещи, продаваемые на различных аукционах, могут иметь одинаковые номера лотов. В книгах фирмы делается запись о каждом аукционе. Там отмечаются дата, место и время его проведения, а также специфика (например, выставляются картины, написанные маслом и не позднее 1950 г.). Заносятся также сведения о каждом продаваемом предмете: аукцион, на который он заявлен, номер лота, продавец, стартовая цена, отправная цена и краткое словесное описание. Продавцу разрешается выставлять любое количество вещей, а покупатель имеет право приобретать любое количество вещей. Одно и то же лицо или фирма может выступать и как продавец, и как покупатель. После аукциона служащие фирмы, проводящей аукционы, записывают фактическую цену, уплаченную за проданный предмет, и фиксируют данные покупателя.

5. База данных студенческой библиотеки.

Разработать информационную систему обслуживания библиотеки, которая содержит следующую информацию: названия книг, ФИО авторов, наименования издательств, год издания, количество страниц, состояние книги, количество иллюстраций, стоимость, название филиала библиотеки: учебной литературы/ научной литературы/ читального зала, в которых находится книга, количество имеющихся в библиотеке экземпляров конкретной книги, количество студентов, которым выдавалась конкретная книга, названия факультетов, в учебном процессе которых используется указанная книга. У студентов имеются читательские билеты, в которых каждый семестр библиотекарь зала учебной литературы делает отметку о проверке отсутствия задолженности по книгам.

6. База данных по учету успеваемости студентов.

База данных должна содержать данные о контингенте студентов (фамилия, имя, отчество, год поступления, форма обучения (дневная/вечерняя/заочная), номер и название группы); об учебном плане (название специальности, дисциплина, семестр, количество отводимых на дисциплину часов, форма отчетности (экзамен/зачет)). Учебная группа числится закрепленной за кафедрой, конкретная дисциплина у конкретной группы также числится за кафедрой согласно учебному плану. В течение любого семестра студенты сдают лабораторные работы и занимаются практическими занятиями по различным предметам из учебного плана. В базе отмечается допуск, выполнение, защита лабораторной работы или отметка на практическом занятии. Также учитывается посещаемость студентов. Допуском до выполнения лабораторной работы – сформированный заранее отчет о предстоящей работе. Выполнение работы – это демонстрация работоспособности программы на языке высокого уровня, предоставление верного решения задачи преподавателю. Защита лабораторной работы состоит в выполнении дополнительного задания, более сложного, требующего углубленные знания в данной области. Все это отмечается в журнале успеваемости по данному предмету в некой группе в текущем семестре. Также требуется возможность предоставления информации для студентов об их задолженностях.

7. База данных музыкального магазина.

Таблицы базы данных содержат информацию о музыкантах, музыкальных произведениях и обстоятельствах их исполнения. Несколько музыкантов, образующих единый коллектив, называются ансамблем. Это может быть классический оркестр, джазовая группа, квартет, квинтет и т.д. К музыкантам причисляют исполнителей (играющих на одном или нескольких инструментах), композиторов, дирижеров и руководителей ансамблей. Кроме того, в базе данных хранится информация о пластинках, которыми магазин торгует. Музыканты записывают музыкальные произведения, которые идентифицируются отдельным номером. На пластинке может быть записано несколько исполнений одного и того же произведения, для каждого из них в базе заведена отдельная запись. Когда выходит новая пластинка, регистрируется название выпускающей ее компании, а также адрес оптовой фирмы, у которой магазин может приобрести эту пластинку. Не исключено, что компания производитель занимается и оптовой продажей своих пластинок. Магазин фиксирует текущие оптовые и розничные цены на каждую пластинку, дату ее выпуска, количество экземпляров, проданных за прошлый год и в нынешнем году, а также число еще не распроданных пластинок.

8. База данных Банковские вклады.

Клиентам банка предлагается размещать денежные средства на срочные банковские вклады различных видов. Клиентом считается человек, открывший любой вклад в банке. Клиент может помещать средства на несколько своих счетов. Клиент осуществляет перемещения средств со вкладов и счетов на вклады и счета других клиентов, а также может осуществлять внутренние перемещения денежных средств. Сотрудники банка периодически обзванивают клиентом для проведения опросом по качеству обслуживания, а также для предложения специальных условий кредитования. Каждый звонок, его тема, результат, сотрудник фиксируется в базе данных. Кредит, который будет оформлен после телефонного звонка должен иметь отметку о телефонном звонке, который послужил причиной взятия кредита. Клиент может оформить кредит, заказать свою кредитную историю. Клиент имеет право провести рефинансирование уже имеющихся кредитов.

9. База данных Аудит.

Сотрудники разных категорий (1, 2, 3, высшая) аудиторской фирмы осуществляют проверку предприятий в течение года. Работа сотрудников оплачивается за каждый час в зависимости от категории сотрудника. Необходимо вести базу данных для автоматизации начисления зарплаты сотрудникам. В базе фиксируется вид работы, выполненный сотрудниками в конкретный день на предприятии и количество часов, затраченное на работу. После проделанной работы сотрудники составляют акты о проделанной работе, в которой фиксируют предприятие, итог аудиторской проверки, предложения по усовершенствованию. Предприятие, прошедшее аудиторскую проверку может обратиться в суд в случае если проверка проходила не в соответствии с законодательством РФ, если сотрудники, проводящие проверку, нарушали законодательство РФ, в отчете о проверке указаны неверные данные о предприятии или проходящих в них процессах, и т.д. Чем больше раз в суде участвует сотрудник аудиторской проверки в случае проверки по сомнению в законности их действий, тем больший процент премии урежут у сотрудника при начислении заработной платы за месяц.

10. База данных туристического агентства.

Работники турагентства продают путевки путешествий по разным странам. В каждую страну организуются несколько маршрутов. По каждому маршруту указывается цель путешествия (отдых,

экскурсия, лечение, шоп-тур, обучение и др.). Необходимо разработать базу данных для определения наиболее популярных маршрутов за текущий год, для отслеживания обращений клиентов. В базе данных ведется статистика отзывов и оценок (от 1 до 5 звезд) по каждому маршруту. Постоянные клиенты имеют скидку в зависимости от количества взятых путевок.

в) типовые теоретические вопросы:

1. Виды баз данных.
2. Группы операторов SQL.
3. Операторы DDL.
4. Операторы DML.
5. Операторы DCL.
6. Операторы TCL.
7. Правила Кодда.
8. Информационное правило.
9. Гарантированный доступ к данным.
10. Систематическая поддержка отсутствующих значений .
11. Возможность изменения представлений. .
12. Наличие высокоуровневых операций управления данными.
13. Физическая независимость данных.
14. Логическая независимость данных.
15. Независимость контроля целостности.
16. Независимость от расположения.
17. Согласование языковых уровней.
18. Уровни моделирования.
19. Виды инфологических моделей.
20. Виды даталогических моделей.
21. Физические модели.
22. Проектирование БД. Основные задачи проектирования БД.
23. Проектирование БД. Основные этапы проектирования БД.
24. Распределенные базы данных. Основные моменты. Пример архитектуры.
25. Методы поддержки распределенных данных. Фрагментация.
26. Методы поддержки распределенных данных. Репликация.
27. Репликация. Модели тиражирования.
28. Методы поддержки распределенных данных. Распределенные ограничения целостности.
29. Методы поддержки распределенных данных. Распределенные запросы.
30. Методы поддержки распределенных данных. Распределенные транзакции.
31. Свойства идеальной РБД. Примеры.
32. Современные технологии доступа к данным. Общие сведения.
33. Современные технологии доступа к данным. ODBC.
34. Современные средства проектирования БД. Обзор существующих решений.
35. Проблемы циклических связей в БД. Способы разрешения проблемы.
36. Реализация наследования в БД. Виды наследования.
37. Реализация обычного и взаимоисключающего наследования в БД. Проблемы добавления новых данных.
38. Реализация обычного и взаимоисключающего наследования в БД. Проблемы обновления данных.
39. Реализация обычного и взаимоисключающего наследования в БД. Проблемы удаления данных.
40. Реализация обычного и законченного наследования в БД. Примеры.
41. Реализация взаимоисключающего законченного наследования в БД. Проблемы добавления новых данных.
42. Реализация взаимоисключающего законченного наследования в БД. Проблемы обновления данных.
43. Реализация взаимоисключающего законченного наследования в БД. Проблемы удаления данных.
44. Проблемы рекурсивных связей в однокорневом дереве. Примеры.
45. Проблемы добавления данных для рекурсивных связей в однокорневом дереве. Реализация

- бизнес-правила: «Каждый сотрудник не должен находиться в роли собственного руководителя».
46. Проблемы обновления данных для рекурсивных связей в однокорневом дереве. Реализация бизнес-правила: «Каждый сотрудник не должен находиться в роли собственного руководителя».
47. Проблемы удаления данных для рекурсивных связей в однокорневом дереве. Реализация бизнес-правила: «Каждый сотрудник не должен находиться в роли собственного руководителя».

4.2. Промежуточная аттестация в форме курсовой работы

Целью курсовой работы является:

получение навыков проектирования моделей данных для разных предметных областей в различных нотациях, программирования запросов на диалектах языка SQL, создания и сопровождения баз данных в современных СУБД, а также получение навыков самостоятельной работы при изучении технической документации в области проектирования, программирования и сопровождения БД.

Типовое задание на курсовую работу:

В соответствии с вариантом задания (см. таблицу) провести проектирование базы данных для указанной предметной области в заданной нотации. Разработать сценарий создания базы данных и основных объектов базы данных на заданном языке программирования БД, реализовать полученную структуру в заданной СУБД.

№ варианта	Предметная область	Нотация для проектирования ER-модели	Диалект языка SQL	СУБД
1	Телекоммуникационный центр	П. Чена,	Transact-SQL	Microsoft SQL Server
2	Прокат спортивного инвентаря	Дж. Мартина,	PL/pgSQL	PostgreSQL
3	Выставка-продажа сельскохозяйственной продукции	Ч. Баркера,	PL/SQL	Oracle Database
4	Дневной стационар	Ж.-Р. Абриаля,	Transact-SQL	Microsoft SQL Server
5	Бюро трудоустройства	IDEF1X	PL/pgSQL	PostgreSQL
6	Показ-продажа моделей высокой моды	П. Чена,	PL/SQL	Oracle Database
7	Видеопрокат	Дж. Мартина,	Transact-SQL	Microsoft SQL Server
8	Чемпионат мира по футболу	Ч. Баркера,	PL/pgSQL	PostgreSQL
9	Риэлторская фирма	Ж.-Р. Абриаля,	PL/SQL	Oracle Database
10	Расписание занятий	IDEF1X	Transact-SQL	Microsoft SQL Server
11	Обувная мастерская	П. Чена,	Transact-SQL	Microsoft SQL Server
12	Тренажерный зал	Дж. Мартина,	PL/pgSQL	PostgreSQL
13	Салон красоты	Ч. Баркера,	PL/SQL	Oracle Database
14	Автовокзал	Ж.-Р. Абриаля,	Transact-SQL	Microsoft SQL Server
15	Туристическая фирма	IDEF1X	PL/pgSQL	PostgreSQL
16	Продажа компьютерной техники	П. Чена,	PL/SQL	Oracle Database
17	Автомагазин	Дж. Мартина,	Transact-SQL	Microsoft SQL Server
18	Спортивные товары	Ч. Баркера,	PL/pgSQL	PostgreSQL
19	Аэропорт	Ж.-Р. Абриаля,	PL/SQL	Oracle Database
20	Ресторан	IDEF1X	Transact-SQL	Microsoft SQL

				Server
--	--	--	--	--------

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. Все действия по внесению изменений в БД оформить в виде хранимых процедур.

При выполнении курсовой работы необходимо использовать материал лекций и рекомендуемую учебно-методическую литературу.

Типовые вопросы при защите курсовой работы:

1. Основные возможности современных СУБД. (ПК-11)
2. Преимущества и недостатки работы в СУБД Microsoft SQL Server. (ПК-11)
3. Преимущества и недостатки работы в СУБД Oracle Database. (ПК-11)
4. Преимущества и недостатки работы в СУБД PostgreSQL. (ПК-11)
5. Основные возможности диалекта Transact-SQL от СУБД Microsoft SQL Server. (ПК-11)
6. Процедурное расширение языка PL/SQL в СУБД Oracle Database. (ПК-11)
7. Основные возможности диалекта PL/pgSQL в СУБД PostgreSQL. (ПК-11)
8. Анализ современных СУБД для решения типовых задач баз данных. (ПК-11)

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

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

Кафедра «Государственного, муниципального и корпоративного управления»

МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Б1.В.ДВ.02.01 «Современные технологии баз данных»

Направление подготовки
38.04.04 «Государственное и муниципальное управление»

Профиль – Информационные технологии в государственном и муниципальном
управлении

ОПОП академической магистратуры
«Государственное и муниципальное управление»

Формы обучения – очно-заочная

1. КОНТРОЛЬНЫЕ ВОПРОСЫ ПО ДИСЦИПЛИНЕ

Введение в БД. Реляционная модель данных.

1. Определение ИС, БД, СУБД. Понятие архитектуры клиент-сервер.
2. Модели БД. Типы СУБД. Функции СУБД.
3. Структурная часть РМД. Основные понятия.
4. Целостная часть РМД. Основные понятия.
5. Операции, нарушающие ссылочную целостность.
6. Основные стратегии поддержания ссылочной целостности.
7. Дополнительные стратегии поддержания ссылочной целостности.
8. Правила Кодда.
9. Информационное правило.
10. Гарантированный доступ к данным.
11. Систематическая поддержка отсутствующих значений .
12. Возможность изменения представлений. .
13. Наличие высокоуровневых операций управления данными.
14. Физическая независимость данных.
15. Логическая независимость данных.
16. Независимость контроля целостности.
17. Независимость от расположения.
18. Согласование языковых уровней.

Реляционная алгебра.

19. Операции объединения, пересечения и вычитания. Особенности объединения, пересечения и вычитания отношений.
20. Операции переименования, объединения, пересечения и декартова произведения.
21. Операции выборки и проекции.
22. Операция эквивалентного соединения.
23. Внутреннее соединение в реляционной алгебре.
24. Внешние соединения в реляционной алгебре.
25. Операции проекции и деления.

Язык SQL.

26. Синтаксис оператора SELECT. Пример.
27. Основные разделы языка SQL. Примеры операторов по каждому разделу.
28. Предложение SELECT и FROM оператора SELECT. Выборка. Исключение строк-дубликатов. Построение вычисляемых полей. Пример.
29. Сравнение значений в предложении WHERE. Операции IN, BETWEEN. Пример.
30. Операции LIKE, IS NULL в предложении WHERE. Пример.
31. Использование агрегатных функций в предложениях оператора SELECT. Пример.
32. Группировка с помощью предложения GROUP BY оператора SELECT. Пример.
33. Использование предложений GROUP BY и HAVING оператора SELECT. Пример.
34. Сортировка в операторе SELECT. Использование предложения TOP. Пример.
35. Декартово произведение с помощью оператора SELECT. Соединение с помощью предложения WHERE. Пример.
36. Типы соединений в предложении FROM. Пример.
37. Оператор объединения UNION. Особенности объединения запросов. Пример.
38. Операторы вычитания EXCEPT и пересечения INTERSECT. Особенности вычитания и пересечения запросов. Пример.
39. Подзапросы. Классификация подзапросов. Пример простого скалярного подзапроса.
40. Простые табличные подзапросы. Пример.
41. Сложные табличные подзапросы. Пример.
42. Оператор INSERT INTO. Оператор UPDATE. Пример.
43. Оператор UPDATE. Оператор DELETE. Пример.

Реализация операций реляционной алгебры в языке SQL.

44. Операция объединения в реляционной алгебре и ее реализация в языке SQL.
45. Операция пересечения в реляционной алгебре и ее реализация в языке SQL.

46. Операция вычитания в реляционной алгебре и ее реализация в языке SQL.
47. Операция декартового произведения в реляционной алгебре и ее реализация в языке SQL.
48. Операция выборки в реляционной алгебре и ее реализация в языке SQL.
49. Операция проекции в реляционной алгебре и ее реализация в языке SQL.
50. Операция внутреннего соединения в реляционной алгебре и ее реализация в языке SQL.
51. Операция левого внешнего соединения в реляционной алгебре и ее реализация в языке SQL.
52. Операция правого внешнего соединения в реляционной алгебре и ее реализация в языке SQL.
53. Операция деления в реляционной алгебре и ее реализация в языке SQL.
54. Реализация операции вычитания с помощью подзапросов.
55. Реализация операции пересечения с помощью подзапросов.
56. Реализация операции деления с помощью подзапросов.

СУБД SQL Server.

57. Создание БД в СУБД SQL Server. Пример.
58. Основные объекты БД SQL Server. Определения.
59. Системные БД SQL Server. Файлы БД. Скрипт создания БД. Переключение между БД. Подключение и отключение БД.
60. Операторы определения и использования переменных в Transact-SQL. Табличные переменные. Пример.
61. Условный оператор и оператор цикла в Transact-SQL. Пример.
62. Оператор выбора (простая форма) в Transact-SQL. Пример.
63. Оператор выбора (поисковая форма) в Transact-SQL. Пример.
64. Системные типы данных в СУБД SQL Server.
65. Создание таблиц. Ограничения первичного и внешнего ключа. Пример.
66. Создание таблиц. Ограничение внешнего ключа. Стратегии поддержания ссылочной целостности. Пример.
67. Создание таблиц. Ограничения CHECK, DEFAULT, UNIQUE. Пример.
68. Создание таблиц. Ограничения CHECK, UNIQUE. Задание имен ограничений. Пример.
69. Создание таблиц. Ограничения CHECK, DEFAULT. Задание столбцов-счетчиков. Пример.
70. Способы изменения структуры таблицы. Пример.
71. Функции работы с датой/временем в СУБД SQL Server. Пример.
72. Способы задания первичного ключа. Пример.
73. Способы задания внешнего ключа. Пример.
74. Умолчания. Способы задания умолчаний. Пример.
75. Уникальные значения. Способы задания уникальных значений. Пример.
76. Ограничения на значения. Способы задания ограничений на значения. Пример.
77. Определяемые пользователем типы данных. Пример.
78. Представления. Модифицируемые представления. Пример. Немодифицируемые представления. Пример.
79. Особенности модификации данных через представления. Пример.
80. Хранимые процедуры. Параметры процедур. Оператор выполнения ХП. Пример.
81. Триггеры. Определение DML и DDL триггеров. Пример.
82. AFTER-триггеры. Пример.
83. INSTEAD OF-триггеры. Пример.

Проектирование моделей данных

48. Проектирование БД. Основные задачи проектирования БД.
49. Проектирование БД. Основные этапы проектирования БД.
50. Постановка задачи проектирования БД. Основные способы проектирования ER-моделей.
51. Избыточность данных. Аномалии обновления. Пример.
52. Нормализация отношений. Типы зависимостей между атрибутами. Пример.
53. Нормализация отношений. 1НФ, 2НФ. Пример.
54. Нормализация отношений. 3НФ. Пример.
55. Нормализация отношений. БКНФ. Пример.
56. Нормализация отношений. 4НФ. Пример.
57. Проектирование реляционных БД с помощью ER-метода. Основные понятия.
58. Этапы проектирования реляционных БД с помощью ER-метода.

59. Правила перехода от ER-диаграммы к предварительным отношениям для бинарных связей 1:1 и 1:N.
60. Правила перехода от ER-диаграммы к предварительным отношениям для связей 1:N и N:N.
61. Правила перехода от ER-диаграммы к предварительным отношениям для связей N:N.
62. Правило формирования предварительных отношений при наличии супертипа и подтипов сущностей. Формирование отношений при наличии дополнительных связей между отдельными экземплярами подтипа и супертипа.
63. Правило формирования предварительных отношений при наличии рекурсии.
64. Современные средства проектирования БД. Обзор существующих решений.

Проверка правильности результатов проектирования моделей данных

65. Избыточность данных.
66. Аномалии обновления.
67. Процесс нормализации отношений.
68. Типы зависимостей между атрибутами.
69. Нормализация отношений 1НФ
70. Приведение отношений к 2НФ.
71. Приведение отношений к 3НФ.
72. Приведение отношений к БКНФ.
73. Приведение отношений к 4НФ.
74. Приведение отношений к 5НФ.

Распределенные БД

75. Распределенные базы данных. Основные моменты. Пример архитектуры.
76. Методы поддержки распределенных данных. Фрагментация.
77. Методы поддержки распределенных данных. Репликация.
78. Репликация. Модели тиражирования.
79. Методы поддержки распределенных данных. Распределенные ограничения целостности.
80. Методы поддержки распределенных данных. Распределенные запросы.
81. Методы поддержки распределенных данных. Распределенные транзакции.
82. Свойства идеальной РБД. Примеры.

Реализация сложных связей в базах данных

83. Проблемы циклических связей в БД. Способы разрешения проблемы.
84. Реализация наследования в БД. Виды наследования.
85. Реализация обычного и взаимоисключающего наследования в БД. Проблемы добавления новых данных.
86. Реализация обычного и взаимоисключающего наследования в БД. Проблемы обновления данных.
87. Реализация обычного и взаимоисключающего наследования в БД. Проблемы удаления данных.
88. Реализация обычного и законченного наследования в БД.
89. Реализация взаимоисключающего законченного наследования в БД. Проблемы добавления новых данных.
90. Реализация взаимоисключающего законченного наследования в БД. Проблемы обновления данных.
91. Реализация взаимоисключающего законченного наследования в БД. Проблемы удаления данных.
92. Проблемы рекурсивных связей в однокорневом дереве.
93. Проблемы добавления данных для рекурсивных связей в однокорневом дереве.
94. Проблемы обновления данных для рекурсивных связей в однокорневом дереве.
95. Проблемы удаления данных для рекурсивных связей в однокорневом дереве.

2. МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ВЫПОЛНЕНИЮ ЛАБОРАТОРНЫХ РАБОТ

Выполнение лабораторных работ осуществляется в соответствии с методической литературой:

1. Громов А.Ю. Современные технологии разработки интегрированных информационных систем: учеб. пособие / А.Ю. Громов, Н.Н. Гринченко, Н.В. Шемонаев; Рязан. гос. радиотехн. ун-т. - Рязань, 2015. - 48 с. <https://elibrseu.ru/ebs/download/562>

2. Клиент-серверные приложения баз данных: учеб. пособие / А.В. Благодаров, Н.Н. Гринченко, А.Ю. Громов; Рязан. гос. радиотехн. ун-т. Рязань, 2017. 72 с. <https://elibrseu.ru/ebs/download/2356>

3. Гринченко Н.Н. Инструментальные средства поддержки проектирования баз данных: учеб. пособие / Н.Н. Гринченко, А.Ю. Громов; Рязан. гос. радиотехн. ун-т. - Рязань, 2015. - 48 с. <https://elibrseu.ru/ebs/download/731>

3. МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО НАПИСАНИЮ КУРСОВОЙ РАБОТЫ ПО ДИСЦИПЛИНЕ:

Целью курсовой работы является:

получение навыков проектирования моделей данных для разных предметных областей в различных нотациях, проверки правильности проектирования моделей данных, программирования запросов на диалектах языка SQL, создания и сопровождения баз данных в современных СУБД.

Задание на курсовую работу:

В соответствии с вариантом задания (см. таблицу) провести проектирование базы данных для указанной предметной области в заданной нотации. Разработать сценарий создания базы данных и основных объектов базы данных на заданном языке программирования БД, реализовать полученную структуру в заданной СУБД, создать основные объекты БД, требуемые пользователю.

№ варианта	Предметная область	Нотация для проектирования ER-модели	Диалект языка SQL	СУБД
1	Телекоммуникационный центр	П. Чена,	Transact-SQL	Microsoft SQL Server
2	Прокат спортивного инвентаря	Дж. Мартина,	PL/pgSQL	PostgreSQL
3	Выставка-продажа сельскохозяйственной продукции	Ч. Баркера,	PL/SQL	Oracle Database
4	Дневной стационар	Ж.-П. Абриаля,	Transact-SQL	Microsoft SQL Server
5	Бюро трудоустройства	IDEF1X	PL/pgSQL	PostgreSQL
6	Показ-продажа моделей высокой моды	П. Чена,	PL/SQL	Oracle Database
7	Видеопрокат	Дж. Мартина,	Transact-SQL	Microsoft SQL Server
8	Чемпионат мира по футболу	Ч. Баркера,	PL/pgSQL	PostgreSQL
9	Риэлторская фирма	Ж.-П. Абриаля,	PL/SQL	Oracle Database
10	Расписание занятий	IDEF1X	Transact-SQL	Microsoft SQL Server
11	Обувная мастерская	П. Чена,	Transact-SQL	Microsoft SQL Server
12	Тренажерный зал	Дж. Мартина,	PL/pgSQL	PostgreSQL
13	Салон красоты	Ч. Баркера,	PL/SQL	Oracle Database
14	Автовокзал	Ж.-П. Абриаля,	Transact-SQL	Microsoft SQL Server
15	Туристическая фирма	IDEF1X	PL/pgSQL	PostgreSQL
16	Продажа компьютерной техники	П. Чена,	PL/SQL	Oracle Database
17	Автомагазин	Дж. Мартина,	Transact-SQL	Microsoft SQL

				Server
18	Спортивные товары	Ч. Баркера,	PL/pgSQL	PostgreSQL
19	Аэропорт	Ж.-Р. Абриаля,	PL/SQL	Oracle Database
20	Ресторан	IDEFIX	Transact-SQL	Microsoft SQL Server

Общие требования к работе:

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

Предполагается, что разрабатываемая БД будет представлять собой серверную часть информационной системы с клиент-серверной архитектурой, использующей двухзвенную модель DBS (сервер баз данных).

Провести инфологическое и даталогическое проектирование БД. Проверить все таблицы БД на соответствие нормальной форме Бойса-Кодда. Предусмотреть контроль целостности данных на уровне ограничений, триггеров, правил, умолчаний и хранимых процедур. Все действия по внесению изменений в БД оформить в виде хранимых процедур.

При выполнении курсовой работы необходимо использовать материал лекций и рекомендуемую учебно-методическую литературу.

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. Задание частных ограничений целостности данных.
 - 4.2. Разработка представлений.
 - 4.3. Разработка запросов к БД.
 - 4.4. Разработка ХП, триггеров.

Заключение.

Список используемой литературы.

Приложение: SQL-скрипт для создания БД (должны быть включены все объекты БД – таблицы, представления, ХП и т.п.).

К защите представить:

- работоспособную БД, установленную на компьютере в кафедральной аудитории ;
- пояснительную записку в распечатанном виде;
- электронный каталог, содержащий пояснительную записку, файлы БД, сценарий создания основных объектов БД, сценарий заполнения таблиц.

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. Постановка задачи и технико-экономическое	2 недели

обоснование разработки.	
3. Разработка общей архитектуры.	1 неделя
4. Разработка моделей данных.	3 недели
5. Создание схемы базы данных и заполнение таблиц данными.	1 неделя
6. Создание ограничений, представлений, хранимых процедур, триггеров.	2 недели
7. Оформление пояснительной записки.	1 неделя
8. Сдача курсовой работы на проверку.	1 неделя
9. Защита курсовой работы.	1 неделя

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

Типовые вопросы при защите курсовой работы:

1. Основные возможности современных СУБД.
2. Преимущества и недостатки работы в СУБД Microsoft SQL Server.
3. Преимущества и недостатки работы в СУБД Oracle Database.
4. Преимущества и недостатки работы в СУБД PostgreSQL.
5. Основные возможности диалекта Transact-SQL от СУБД Microsoft SQL Server.
6. Процедурное расширение языка PL/SQL в СУБД Oracle Database.
7. Основные возможности диалекта PL/pgSQL в СУБД PostgreSQL.
8. Анализ современных СУБД для решения типовых задач баз данных.
9. Особенности нотации для проектирования ER-моделей.
10. Особенности нотации П. Чена.
11. Особенности нотации Дж. Мартина.
12. Особенности нотации Ч. Баркера.
13. Особенности нотации Ж.-Р. Абриаля.
14. Особенности нотации IDEF1X.

3. ПРИМЕР ОФОРМЛЕНИЯ НЕКОТОРЫХ РАЗДЕЛОВ ПОЯСНИТЕЛЬНОЙ ЗАПИСКИ К КУРСОВОЙ РАБОТЕ

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.

3.1. 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 (поочерёдно);
- 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.

3.2. 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.

3.2.1. 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.

3.2.2. 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:

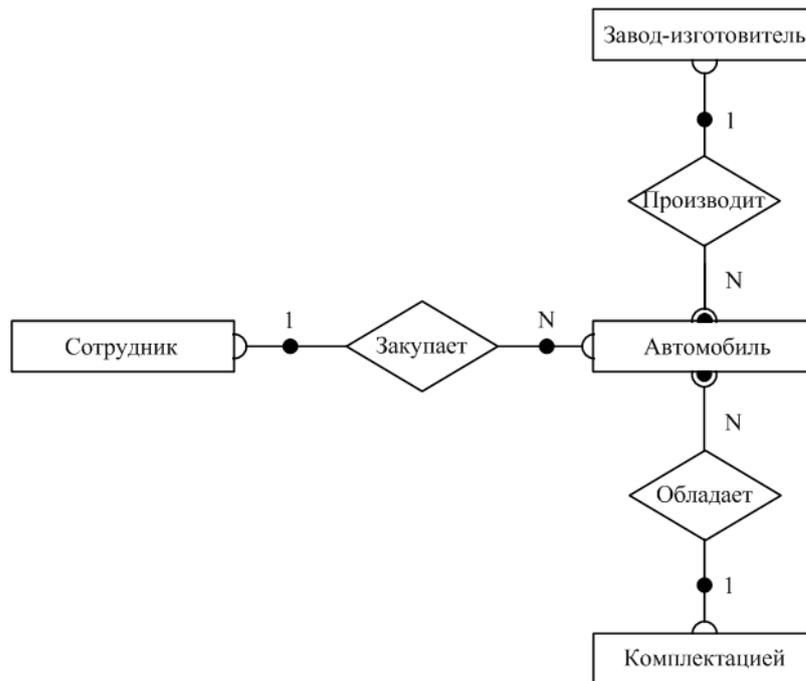


Рисунок 2 – ER-диаграмма связи «сотрудник закупает автомобиль некоторой комплектации у завода-изготовителя»

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:

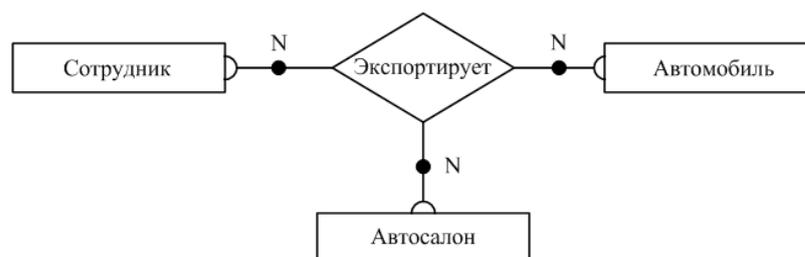


Рисунок 3 – ER-диаграмма связи «Сотрудник экспортирует автомобиль в салон»

Типы связей: многие ко многим (сотрудник экспортирует множество автомобилей во 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:

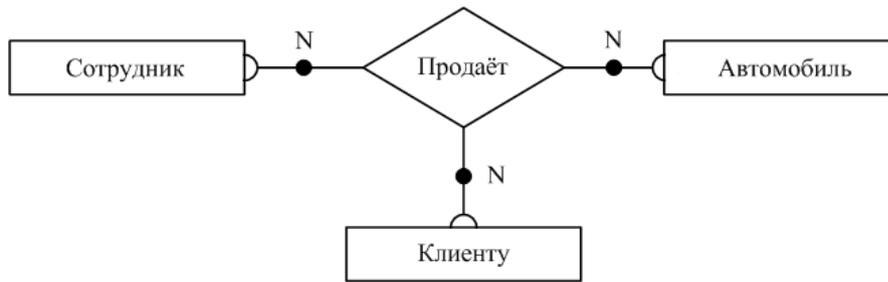


Рисунок 4 – ER-диаграмма связи «Сотрудник продаёт автомобиль клиенту»

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:

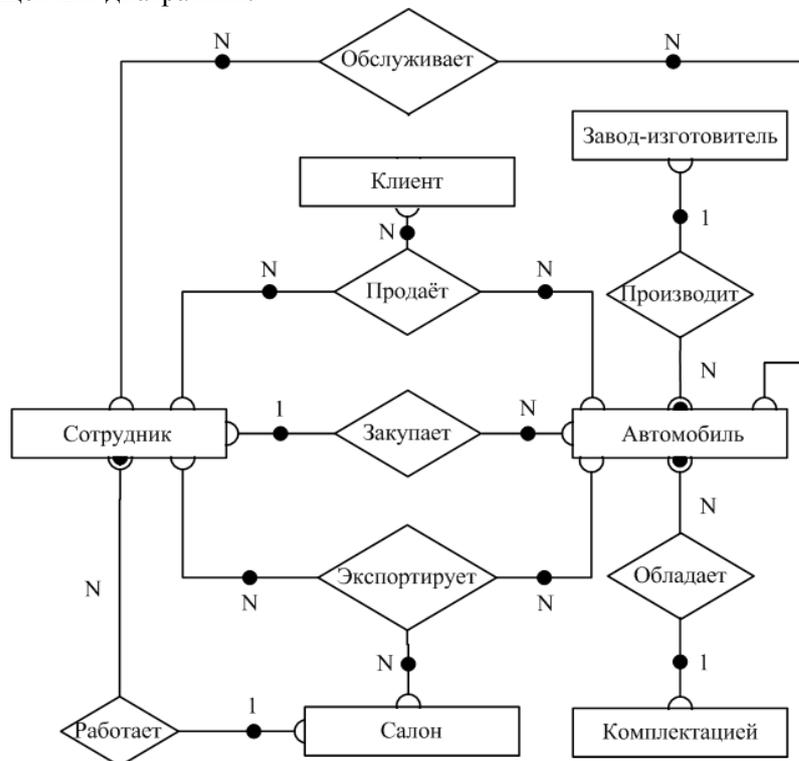


Рисунок 6 – Общая ER-диаграмма

По результатам первых двух этапов проектирования была получена инфологическая модель –

полная ER-диаграмма классов, представляющая собой объединение всех построенных ER-диаграмм для отдельных связей.

3.2.3. 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(ИДАвтомобиля, №Завода, №Комплектации),
Экспорт(ИДАвтомобиля, 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(ИДАвтомобиля,
№Завода, №Комплектации),
Клиент(ФИОКлиента),
Продажа(ИДАвтомобиля, 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(ИДАвтомобиля, №Завода, №Комплектации),
Обслуживание(ИДАвтомобиля, ФИОСотрудника).

3.2.4. 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.

3.2.5. 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(ИДАвтомобиля, №Завода, Модель, №Комплектации);
- 6) Клиент(ФИОКлиента, ТелефонКлиента);
- 7) Закупка(ИДАвтомобиля, ФИОСотрудника, ДатаЗакупки, СтоимостьЗакупки);
- 8) Экспорт(ИДАвтомобиля, ФИОСотрудника, №Офиса, ДатаЭкспорта);
- 9) Продажа(ИДАвтомобиля, ФИОСотрудника, ФИОКлиента, ДатаПродажи, СтоимостьПродажи);
- 10) Обслуживание(ИДАвтомобиля, ФИОСотрудника, ДатаОбслуживания, ПричинаОбращения).

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

3.2.6. 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:

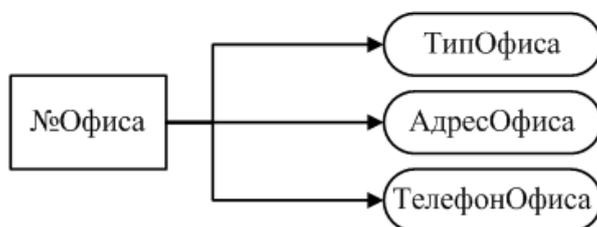


Рисунок 7 – Отношение «Офис»

2) Сотрудник:

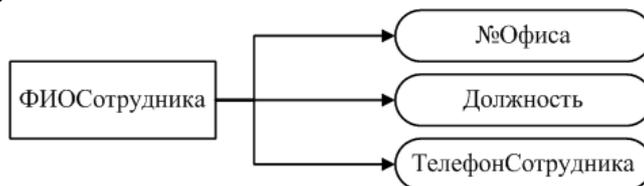


Рисунок 8 – Отношение «Сотрудник»

3) Завод-изготовитель:

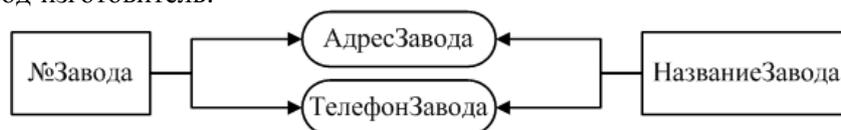


Рисунок 9 – Отношение «Завод-изготовитель»

4) Комплектация:



Рисунок 10 – Отношение «Комплектация»

5) Автомобиль:

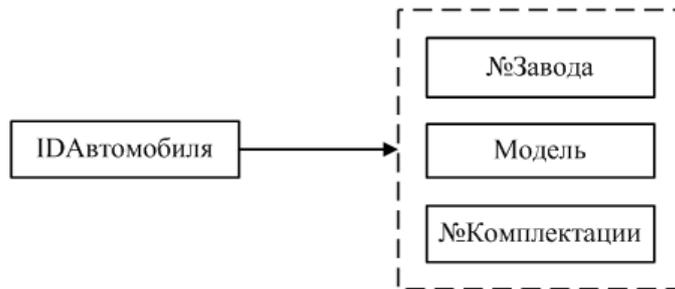


Рисунок 11 – Отношение «Автомобиль»

6) Клиент:



Рисунок 12 – Отношение «Клиент»

7) Закупка

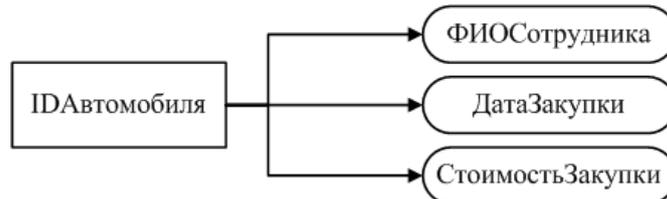


Рисунок 13 – Отношение «Закупка»

8) Экспорт:

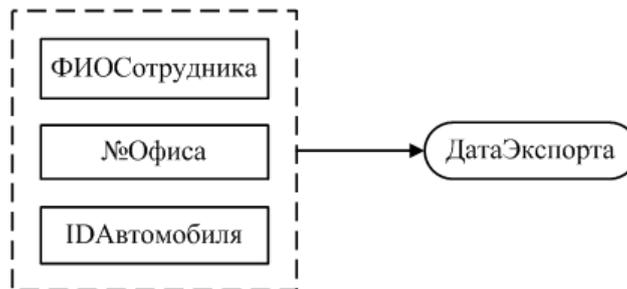


Рисунок 14 – Отношение «Экспорт»

9) Продажа:

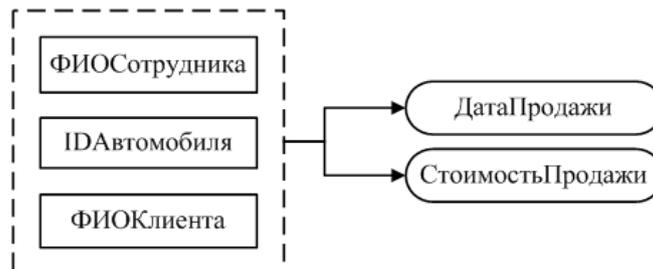


Рисунок 15 – Отношение «Продажа»

10) Обслуживание:

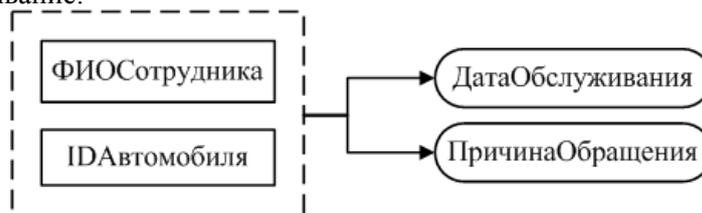


Рисунок 16 – Отношение «Обслуживание»

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.

3.2.7. Пересмотр ER-диаграммы

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

3.2.8. Построение схемы БД

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

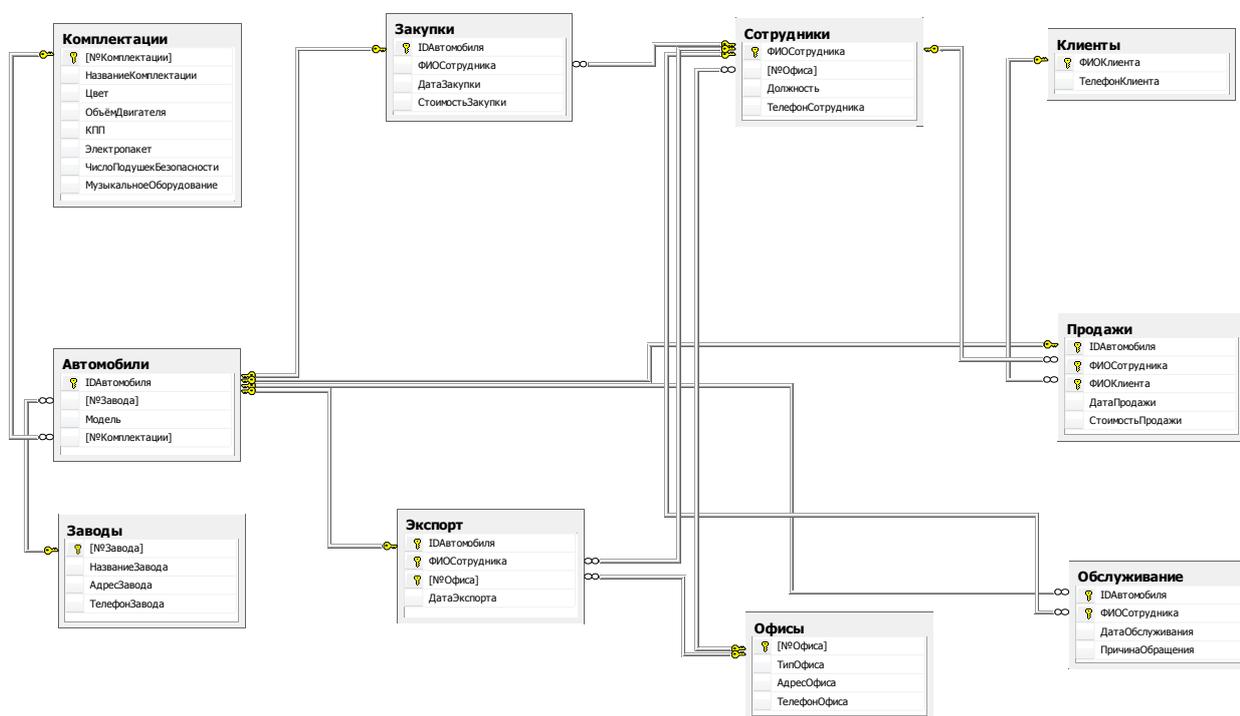


Рисунок 17 – Схема базы данных

3.3. Разработка БД и основных объектов БД.

3.3.1. 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',
    Filename='C:\Avto_Data.mdf',
    Size=3,
    Maxsize=20,
    Filegrowth=1)
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 (ИДАвтомобиля),
constraint FK_Закупки_Автомобили foreign key (ИДАвтомобиля)
references Автомобили(ИДАвтомобиля),
constraint FK_Закупки_Сотрудники foreign key
(ФИОСотрудника) references Сотрудники(ФИОСотрудника))

```

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

```

create table Экспорт(ИДАвтомобиля int unique, ФИОСотрудника
nvarchar(50), [№Офиса] int not null, ДатаЭкспорта datetime not null,
constraint PK_Экспорт primary key (ФИОСотрудника, [№Офиса],
ИДАвтомобиля),
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 (ИДАвтомобиля)
references Автомобили(ИДАвтомобиля))

```

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

```

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

```

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

```

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

```

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

3.3.2. 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'
```

Надобности в разработке большего количества правил нет.

3.3.3. Разработка умолчаний

Был разработан ряд умолчаний, в соответствии с которыми будут задаваться значения неопределённым величинам. Эти значения определялись по принципу наиболее популярного варианта значения поля.

1. По умолчанию тип офиса будет приниматься равным 'Торгово-сервисное представительство', поскольку центральный офис всего один:

```
create default ТипОфисаПоУмолч  
as 'Торгово-сервисное представительство'  
go  
exec sp_bindefault 'ТипОфисаПоУмолч', 'Офисы.ТипОфиса'  
go
```

2. Должность сотрудника по умолчанию принимается равной 'Менеджер', поскольку это преобладающая профессия работников предприятия

```
create default ДолжностьПоУмолч as 'Менеджер'  
go  
exec sp_bindefault 'ДолжностьПоУмолч', 'Сотрудники.Должность'  
go
```

3. Цвет по умолчанию – Серебристый:

```
create default ЦветПоУмолч as 'Серебристый'  
go  
exec sp_bindefault 'ЦветПоУмолч', 'Комплектации.Цвет'  
go
```

4. По умолчанию объём двигателя принимается равным 1.5 литра:

```
create default ОбъёмДвигателяПоУмолч as 1.5  
go  
exec sp_bindefault 'ОбъёмДвигателяПоУмолч',  
  'Комплектации.ОбъёмДвигателя'  
go
```

5. КПП по умолчанию – Механическая (как наиболее распространённая):

```
create default КПППоУмолч as 'Механическая'  
go  
exec sp_bindefault 'КПППоУмолч', 'Комплектации.КПП'  
go
```

6. Электропакет по умолчанию Частичный:

```
create default ЭлектропакетПоУмолч as 'Частичный'
go
exec sp_bindefault 'ЭлектропакетПоУмолч',
'Комплектации.Электропакет'
go
```

7. По умолчанию число подушек безопасности равно 2:

```
create default ЧислоПодушекБезопасностиПоУмолч as 2
go
exec sp_bindefault 'ЧислоПодушекБезопасностиПоУмолч',
'Комплектации.ЧислоПодушекБезопасности'
go
```

8. По умолчанию устанавливается музыкальное оборудование фирмы Mystery

```
create default МузыкальноеОборудованиеПоУмолч as 'Mystery'
go
exec sp_bindefault 'МузыкальноеОборудованиеПоУмолч',
'Комплектации.МузыкальноеОборудование'
go
```

9. По умолчанию завод-изготовитель - ВАЗ (№1) так как основную массу закупаемых автомобилей составляет его продукция:

```
create default НомерЗаводаПоУмолч as 1
go
exec sp_bindefault 'НомерЗаводаПоУмолч', 'Автомобили.№Завода'
go
```

10. По умолчанию принимается комплектация Средняя (№2):

```
create default НомерКомплектацииПоУмолч as 2
go
exec sp_bindefault 'НомерКомплектацииПоУмолч',
'Автомобили.№Комплектации'
go
```

11. Стоимость автомобиля по умолчанию принимается равной 300000 р:

```
create default СтоимостьПоУмолч as 300000
go
exec sp_bindefault 'СтоимостьПоУмолч', 'Закупки.СтоимостьЗакупки'
exec sp_bindefault 'СтоимостьПоУмолч', 'Продажи.СтоимостьПродажи'
go
```

12. По умолчанию автомобили экспортируются во 2й офис (в Германию):

```
create default НомерОфисаПоУмолч as 2
go
exec sp_bindefault 'НомерОфисаПоУмолч', 'Экспорт.№Офиса'
exec sp_bindefault 'НомерОфисаПоУмолч', 'Сотрудники.№Офиса'
go
```

3.3.4. Разработка представлений

Представления в базе данных создавались с учётом их практической надобности. Некоторые из них использовались для облегчения процесса разработки БД.

1. Создание представления, позволяющего просмотреть список менеджеров:

```
create view Менеджеры
as select ФИОСотрудника, [№Офиса], Должность from Сотрудники
where Должность='Менеджер'
```

2. Создание представления, позволяющего просмотреть список автомехаников:

```
create view Автомеханики
as select ФИОСотрудника, [№Офиса], Должность from Сотрудники
where Должность='Автомеханик'
```

3. Создание представления, позволяющего просмотреть список непроданных автомобилей:

```
create view НепроданныеАвтомобили
as select * from Автомобили where IDАвтомобиля not in (select
IDАвтомобиля from Продажи)
```

4. Создание представления, позволяющего просмотреть список автомобилей, которым не предоставлялось сервисное обслуживание:

```
create view НеобслуженныеАвтомобили
as select * from Автомобили where (IDАвтомобиля in (select
IDАвтомобиля from Продажи)) and (IDАвтомобиля not in (select IDАвтомобиля
from Обслуживание))
```

5. Создание представления, позволяющего просмотреть прибыль от продажи каждого автомобиля:

```
create view ПрибыльОтПродажи
as select Продажи.IDАвтомобиля, Продажи.ФИОСотрудника,
Закупки.СтоимостьЗакупки, Продажи.СтоимостьПродажи,
Прибыль=(СтоимостьПродажи-СтоимостьЗакупки) from
(Закупки inner join Автомобили
on Закупки.IDАвтомобиля=Автомобили.IDАвтомобиля inner join Продажи
on Автомобили.IDАвтомобиля=Продажи.IDАвтомобиля)
```

6. Создание представления, позволяющего просмотреть прибыль, полученную каждым менеджером от продаж автомобилей:

```
create view ПрибыльПоМенеджерам
as select distinct Менеджеры.ФИОСотрудника,
ПрибыльСотрудника=(select sum (ПрибыльОтПродажи.Прибыль) from
ПрибыльОтПродажи
where
Менеджеры.ФИОСотрудника=ПрибыльОтПродажи.ФИОСотрудника group by
ФИОСотрудника)
from Менеджеры left join ПрибыльОтПродажи on
Менеджеры.ФИОСотрудника=ПрибыльОтПродажи.ФИОСотрудника
```

3.3.5. Разработка хранимых процедур

Разработанные хранимые процедуры можно подразделить на четыре основных типа: для добавления строк в базовые таблицы, для изменения данных базовых таблиц, для выборки данных и для удаления данных. Для каждой процедуры приведено по два запроса, первый из которых приводит к ошибке, а второй – верен.

1) Процедуры добавления данных в таблицы-справочники:

Добавление данных в таблицу Автомобили:

```
drop proc Добавить_Автомобиль
go
create proc Добавить_Автомобиль (@IDАвтомобиля int, @НомерЗавода
int, @Модель nvarchar(20), @НомерКомплектации int)
as if not exists(select * from Автомобили where
IDАвтомобиля=@IDАвтомобиля)
```

```

        insert into Автомобили
        values (@IDАвтомобиля, @НомерЗавода, @Модель,
@НомерКомплектации)
        else print('Автомобиль с таким ID уже существует')
        go

exec Добавить_Автомобиль 1, 1, '2114', 2
exec Добавить_Автомобиль 21, 5, 'Мобиль', 3

```

Добавление данных в таблицу Заводы:

```

drop proc Добавить_Завод
go
create proc Добавить_Завод(@НомерЗавода int, @НазваниеЗавода
nvarchar(10), @АдресЗавода nvarchar(50),
@ТелефонЗавода nvarchar(20))
as if not exists(select * from Заводы
where ([№Завода]=@НомерЗавода) or (НазваниеЗавода=@НазваниеЗавода))
insert into Заводы
values (@НазваниеЗавода, @АдресЗавода, @ТелефонЗавода)
else print('Информация о заводе с таким номером или названием
уже существует')
go

exec Добавить_Завод 1, 'ВАЗ', 'Тольятти, Автомобилистов, 1', '55-
66-77'
exec Добавить_Завод 6, 'Урал', 'Екатеринбург, Полевая, 6', '543-21-
00'

```

Добавление данных в таблицу Клиенты:

```

drop proc Добавить_Клиента
go
create proc Добавить_Клиента(@ФИОКлиента nvarchar(50),
@ТелефонКлиента nvarchar(20))
as if not exists(select * from Клиенты where
ФИОКлиента=@ФИОКлиента)
insert into Клиенты
values (@ФИОКлиента, @ТелефонКлиента)
else print('Клиент с таким ФИО уже существует')
go

exec Добавить_Клиента 'Блюм Б. Б.', null
exec Добавить_Клиента 'Стрейзанд С. С.', '89567884409'

```

Добавление данных в таблицу Комплектации:

```

drop proc Добавить_Комплектацию
go
create proc Добавить_Комплектацию(@НомерКомплектации int,
@НазваниеКомплектации nvarchar(20), @Цвет nvarchar(15), @ОбъёмДвигателя
real, @КПП nvarchar(20), @Электропакет nvarchar(20),
@ЧислоПодушекБезопасности int, @МузыкальноеОборудование nvarchar(20))
as if not exists(select * from Комплектации
where (@НомерКомплектации=[№Комплектации] or
НазваниеКомплектации=@НазваниеКомплектации))

```

```

insert into Комплектации
values (@НазваниеКомплектации, @Цвет, @ОбъёмДвигателя,
@КПП, @Электропакет,
@ЧислоПодушекБезопасности,
@МузыкальноеОборудование)
else print('Комплектация с таким номером или названием уже
существует')
go

exec Добавить_Комплектацию 1, 'Абракадабра', 'Синий', 2,
'Механическая', 'Полный', 3, 'AIWA'
exec Добавить_Комплектацию 4, 'Люкс', 'Чёрный', 2.1,
'Автоматическая', 'Полный', 7, 'BeatsAudio'

```

Добавление данных в таблицу Офисы:

```

create proc Добавить_Офис(@НомерОфиса int, @ТипОфиса nvarchar(50),
@АдресОфиса nvarchar(50), @ТелефонОфиса nvarchar(20))
as if not exists(select * from Офисы where
[№Офиса]=@НомерОфиса)
insert into Офисы
values(@ТипОфиса, @АдресОфиса, @ТелефонОфиса)
else print('Офис с таким номером уже существует')
go

exec Добавить_Офис 2, 'Торгово-сервисное представительство',
'Германия, Мюнхен, Зеештрассе, 16', '444-55-66'
exec Добавить_Офис 8, 'Торгово-сервисное представительство',
'Швеция, Копенгаген, Карла Маркса, 15', '777-55-66'

```

Добавление данных в таблицу Сотрудники:

```

create proc Добавить_Сотрудника(@ФИОСотрудника nvarchar(50),
@НомерОфиса int, @Должность nvarchar(50),
@ТелефонСотрудника
nvarchar(20))
as if not exists(select * from Сотрудники where
ФИОСотрудника=@ФИОСотрудника)
insert into Сотрудники
values(@ФИОСотрудника, @НомерОфиса, @Должность,
@ТелефонСотрудника)
else print('Сотрудник с таким ФИО уже существует')
go

exec Добавить_Сотрудника 'Абрамов А. А.', 5, 'Автомеханик',
NULL
exec Добавить_Сотрудника 'Абрамович А. А.', 7, 'Автомеханик',
'89038372639'

```

2). Процедуры модификации данных в таблицах-справочниках:

Изменение требований к комплектациям автомобилей:

Изменение количества подушек безопасности:

```

create proc Изменить_Число_Подушек(@Штук int)

```

```

as update Комплектации set
ЧислоПодушекБезопасности=ЧислоПодушекБезопасности+@Штук
go

```

```

exec Изменить_Число_Подушек 1
exec Изменить_Число_Подушек -1

```

Изменение объёма двигателя:

```

create proc Изменить_Объём_Двигателя(@Литров real)
as update Комплектации set
ОбъёмДвигателя=ОбъёмДвигателя+@Литров
go

```

```

exec Изменить_Объём_Двигателя 1
exec Изменить_Объём_Двигателя -1

```

Данный тип процедур нерационален в представленной базе данных, так как она имеет дело с уже произведёнными автомобилями, характеристики которых изменить уже невозможно. Данные запросы разобраны для теоретического примера.

3) Процедуры выборки данных

Определение фамилии менеджера, закупившего автомобиль по IDАвтомобиля:

```

create proc Закупивший_Сотрудник(@IDАвтомобиля int)
as
declare @ФИОСотрудника nvarchar(50)
select @ФИОСотрудника = ФИОСотрудника from Закупки where
Закупки.IDАвтомобиля=@IDАвтомобиля
print 'Автомобиль с ID равным '+convert(char(2),
@IDАвтомобиля)+' закупил '+@ФИОСотрудника
go

```

```

exec Закупивший_Сотрудник 12

```

Слежение за движением автомобиля:

```

create proc Где_Автомобиль(@IDАвтомобиля int)
as
if not exists (select * from Закупки where
Закупки.IDАвтомобиля=@IDАвтомобиля)
print 'Автомобиль с ID равным '+convert(char(2),
@IDАвтомобиля)+' не был закуплен'
else
begin
if not exists (select * from Продажи where
Продажи.IDАвтомобиля=@IDАвтомобиля)
print 'Автомобиль с ID равным '+convert(char(2),
@IDАвтомобиля)+' был закуплен, но не был продан'
else
begin
if not exists (select * from Обслуживание where
Обслуживание.IDАвтомобиля=@IDАвтомобиля)
print 'Автомобиль с ID равным '+convert(char(2),
@IDАвтомобиля)+' был закуплен, продан, но не был обслужен'

```

```

                else
                    print 'Автомобиль с ID равным
'+convert(char(2), @IDАвтомобиля)+' был закуплен, продан и был обслужен'
                end
            end
    end

```

exec Где_Автомобиль 20

Расчёт прибыли фирмы в месяце, соответствующем введённой дате:

```

drop proc Месячная_Прибыль
create proc Месячная_Прибыль (@Дата datetime)
    as
    declare @Прибыль money
    select @Прибыль=0
    select @Прибыль=sum(СтоимостьПродажи-СтоимостьЗакупки) from
Закупки inner join Продажи
        on Закупки.IDАвтомобиля=Продажи.IDАвтомобиля where
month(ДатаПродажи)=month(@Дата) and
        year(ДатаПродажи)=year(@Дата)
    if @Прибыль > 0
        print 'В '+convert(char(2), month(@Дата))+'-м месяце
'+convert(char(4), year(@Дата))+'-го года прибыль составила '
            +convert(char(10), @Прибыль)+' рублей'
        else print 'В '+convert(char(2), month(@Дата))+'-м месяце
'+convert(char(4), year(@Дата))+'-го года прибыли не было'

exec Месячная_Прибыль '12.11.11'

```

4) Процедуры удаления данных из таблиц-справочников:

Удаление сотрудника по его ФИО:

```

create proc Удалить_Сотрудника (@ФИОСотрудника nvarchar(50))
    as
    delete from Сотрудники where ФИОСотрудника=@ФИОСотрудника
go

```

Удаление завода по его номеру:

```

create proc Удалить_Завод (@НомерЗавода int)
    as
    delete from Заводы where [№Завода]=@НомерЗавода
go

```

Удаление клиента по его ФИО:

```

create proc Удалить_Клиента (@ФИОКлиента nvarchar(50))
    as
    delete from Клиенты where ФИОКлиента=@ФИОКлиента
go

```

Удаление комплектации по её номеру:

```

create proc Удалить_Комплектацию (@НомерКомплектации int)
as
delete from Комплектации where
[№Комплектации]=@НомерКомплектации
go

```

Удаление офиса по его номеру:

```

create proc Удалить_Офис (@НомерОфиса int)
as
delete from Офисы where [№Офиса]=@НомерОфиса
go

```

Удаление автомобиля по его ID:

```

create proc Удалить_Автомобиль (@IDАвтомобиля int)
as
delete from Автомобили where IDАвтомобиля=@IDАвтомобиля
go

```

3.3.6. Разработка триггеров

Для контроля за соблюдением бизнес-правил, описанных ранее, к таблицам базы данных были подкреплены следующие триггеры:

1. Создание триггера, контролирующего то, чтобы закупки автомобилей осуществлялись только сотрудниками центрального офиса, имеющими должность Менеджер:

```

create trigger Контроль_Закупок on Закупки
instead of insert, update
as
declare @IDАвтомобиля int, @ФИОСотрудника nvarchar(50),
@ДатаЗакупки datetime, @СтоимостьЗакупки money
select @IDАвтомобиля=IDАвтомобиля,
@ФИОСотрудника=ФИОСотрудника, @ДатаЗакупки=ДатаЗакупки,
@СтоимостьЗакупки=СтоимостьЗакупки from inserted
where (@ФИОСотрудника in (select ФИОСотрудника from Сотрудники
where [№Офиса]=1 and Должность='Менеджер'))
and @IDАвтомобиля in (select IDАвтомобиля from
Автомобили)
insert into Закупки values (@IDАвтомобиля, @ФИОСотрудника,
@ДатаЗакупки, @СтоимостьЗакупки)
else print 'Закупка невозможна'

insert into Закупки values (21, 'Сидоров С. С.', '11.11.2011',
400000)

```

2. Создание триггера, контролирующего то, чтобы дата экспорта была позже даты закупки, чтобы экспорт производился только сотрудниками центрального офиса, имеющими должность Менеджер и чтобы автомобиль не мог быть экспортирован в центральный офис:

```

create trigger Контроль_Экспорта on Экспорт
instead of insert, update
as
declare @IDАвтомобиля int, @ФИОСотрудника nvarchar(50),
@НомерОфиса int, @ДатаЭкспорта datetime, @ДатаЗакупки datetime
select @IDАвтомобиля=IDАвтомобиля,

```

```

@ФИОСотрудника=ФИОСотрудника, @НомерОфиса=[№Офиса],
        @ДатаЭкспорта=ДатаЭкспорта from inserted
select      @ДатаЗакупки=ДатаЗакупки      from      Закупки      where
Закупки.IDАвтомобиля=@IDАвтомобиля
        if (@IDАвтомобиля in (select IDАвтомобиля from Закупки)) and
(@ДатаЭкспорта >= @ДатаЗакупки) and
        (@ФИОСотрудника in (select ФИОСотрудника from Сотрудники
where [№Офиса]=1 and Должность='Менеджер')) and
        (@НомерОфиса <>1)
insert into Экспорт values (@IDАвтомобиля, @ФИОСотрудника,
@НомерОфиса, @ДатаЭкспорта)
        else print 'Экспорт невозможен'
go

insert into Экспорт values (20, 'Сидоров С. С.', 7, '11.09.11')
insert into Экспорт values (20, 'Кузьмин К. К.', 7, '11.11.11')
insert into Экспорт values (20, 'Сидоров С. С.', 1, '11.11.11')
insert into Экспорт values (20, 'Сидоров С. С.', 7, '11.11.11')

```

3. Создание триггера, контролирующего то, чтобы дата экспорта была позже даты закупки и чтобы автомобиль был продан в том офисе, куда был экспортирован только сотрудником, имеющим должность Менеджер:

```

create trigger Контроль_Продаж on Продажи
instead of insert, update
as
declare @IDАвтомобиля int, @ФИОСотрудника nvarchar(50),
@ФИОКлиента nvarchar(50), @ДатаПродажи datetime,
@СтоимостьПродажи money, @ДатаЭкспорта datetime
select      @IDАвтомобиля=IDАвтомобиля,
@ФИОСотрудника=ФИОСотрудника, @ФИОКлиента=ФИОКлиента,
@ДатаПродажи=ДатаПродажи,
@СтоимостьПродажи=СтоимостьПродажи from inserted
select      @ДатаЭкспорта=ДатаЭкспорта      from      Экспорт      where
Экспорт.IDАвтомобиля=@IDАвтомобиля
        if (@IDАвтомобиля in (select IDАвтомобиля from Экспорт)) and
(@ДатаПродажи >= @ДатаЭкспорта) and
        (@ФИОСотрудника in (select ФИОСотрудника from Сотрудники
where Сотрудники.[№Офиса] in
        (select      [№Офиса]      from      Экспорт      where
IDАвтомобиля=@IDАвтомобиля)) and
        (@ФИОСотрудника in (select ФИОСотрудника from
Сотрудники where [№Офиса]<>1 and Должность='Менеджер'))
insert into Продажи values (@IDАвтомобиля, @ФИОСотрудника,
@ФИОКлиента, @ДатаПродажи, @СтоимостьПродажи)
        else print 'Продажа невозможна'
go

insert into Продажи values (19, 'Володин В. В.', 'Бергер Б. Б.',
'11.09.11', 500000)
insert into Продажи values (19, 'Чугунов Ч. Ч.', 'Бергер Б. Б.',
'11.13.11', 500000)
insert into Продажи values (19, 'Абрамов А. А.', 'Бергер Б. Б.',
'11.13.11', 500000)
insert into Продажи values (19, 'Сидоров С. С.', 'Бергер Б. Б.',
'11.13.11', 500000)
insert into Продажи values (19, 'Володин В. В.', 'Бергер Б. Б.',
'11.13.11', 500000)

```

4. Создание триггера, контролирующего то, чтобы дата сервисного обслуживания была позже даты продажи и чтобы сервисное обслуживание осуществляли только сотрудники, имеющие должность Автомеханик:

```
create trigger Контроль_Обслуживания on Обслуживание
instead of insert, update
as
declare @IDАвтомобиля int, @ФИОСотрудника nvarchar(50),
@ДатаОбслуживания datetime, @ПричинаОбращения nvarchar(50),
@ДатаПродажи datetime
select @IDАвтомобиля=IDАвтомобиля,
@ФИОСотрудника=ФИОСотрудника, @ДатаОбслуживания=ДатаОбслуживания,
@ПричинаОбращения=ПричинаОбращения from inserted
select @ДатаПродажи=ДатаПродажи from Продажи where
Продажи.IDАвтомобиля=@IDАвтомобиля
if (@IDАвтомобиля in (select IDАвтомобиля from Продажи) and
(@ДатаОбслуживания >= @ДатаПродажи) and
(@ФИОСотрудника in (select ФИОСотрудника
from Сотрудники where [№Офиса]<>1 and Должность='Автомеханик'))
insert into Обслуживание values (@IDАвтомобиля,
@ФИОСотрудника, @ДатаОбслуживания, @ПричинаОбращения)
else print 'Обслуживание невозможно'
go

insert into Обслуживание values (19, 'Юшин Ю. Ю.', '11.25.11',
'Покраска')
insert into Обслуживание values (9, 'Юшин Ю. Ю.', '11.01.11',
'Покраска')
insert into Обслуживание values (9, 'Кузьмин К. К.', '11.25.11',
'Покраска')
insert into Обслуживание values (9, 'Юшин Ю. Ю.', '11.25.11',
'Покраска')
```