

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Хоружий Людмила Ивановна
Должность: Директор института экономики и управления АПК
Дата подписания: 15.07.2023 19:31:30
Уникальный программный ключ:
1e90b132d9b04dce67585160b015dddf2cb1e6a9



УТВЕРЖДАЮ:

Директор института экономики и
управления АПК

Л.И. Хоружий

2022 г.

Лист актуализации рабочей программы дисциплины Б1.О.16 «Базы данных»

для подготовки бакалавров
Направление: 09.03.03 Прикладная информатика
Направленность: Прикладная информатика в экономике
Форма обучения: очная
Год начала подготовки: 2019
Курс 2
Семестр 3, 4

А) В рабочую программу вносятся изменения: изменился шифр дисциплины на Б1.О.20.02. Программа актуализирована для 2022 г. начала подготовки.

Б) Программа будет распространена при организации учебного процесса на направленность (профиль) «Системы искусственного интеллекта».

Разработчики: Худякова Е.В., д.э.н., профессор,

Ханжиян К.И.,

Белоярская Т.С.

(ФИО, ученая степень, ученое звание)

«29» августа 2022 г.

Рабочая программа пересмотрена и одобрена на заседании кафедры прикладной информатики протокол № 1 от «29» августа 2022 г.

Заведующий кафедрой: Худякова Е.В., д.э.н., профессор

Лист актуализации принят на хранение:

Заведующий выпускающей кафедрой

прикладной информатики

Худякова Е.В., д.э.н., профессор

«29» августа 2022 г.



МИНИСТЕРСТВО СЕЛЬСКОГО ХОЗЯЙСТВА РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ АГРАРНЫЙ УНИВЕРСИТЕТ –
МСХА имени К. А. ТИМИРЯЗЕВА»
(ФГБОУ ВО РГАУ - МСХА имени К. А. Тимирязева)

Институт экономики и управления АПК
Кафедра прикладной информатики

УТВЕРЖДАЮ:

И.о. директора института экономики и
управления АПК

Л.И. Хоружий
“ 26 ” Февраль 2020 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Б1.О.16 «Базы данных»

для подготовки бакалавров

ФГОС ВО

Направление: 09.03.03 «Прикладная информатика»

Направленность: «Прикладная информатика в экономике»

Курс: 2

Семестр: 3,4

Форма обучения: очная

Год начала подготовки: 2019

Регистрационный номер _____

Москва, 2020

Разработчики: Худякова Е.В., д.э.н., проф., Ханжиян К.И., Белоярская Т.С.

«09» 11 2019 г.

Рецензент: Харитонов А.Е., к.э.н., доцент

(ФГО, ученым степень, учебное звание)

«05» 12 2019 г.

Программа составлена в соответствии с требованиями ФГОС ВО по направлению подготовки 09.03.03 «Прикладная информатика» и учебного плана по данному направлению.

Программа обсуждена на заседании кафедры прикладной информатики протокол № 4 от «08 12 2019 г.

Зав. кафедрой: Худякова Е.В., д.э.н., профессор

(ФГО, ученым степень, учебное звание)

(подпись)

«09» 12 2019 г.

Согласовано:

Председатель учебно-методической комиссии института экономики и управления АПК:

Корольков А.Ф., к.э.н., доцент

(ФГО, ученым степень, учебное звание)

№ «05» 02 2019 г.

Заведующий выпускающей кафедрой прикладной информатики:

Худякова Е.В., д.э.н., профессор

(ФГО, ученым степень, учебное звание)

(подпись)

«09» 12 2019 г.

Зав. отдела комплектования ЦНБ

(подпись)

Бумажный экземпляр РПД, копии электронных вариантов РПД и оценочных материалов получены:

Методический отдел УМУ

«__» ____ 20 г.

СОДЕРЖАНИЕ

АННОТАЦИЯ	4
1. ЦЕЛЬ ОСВОЕНИЯ ДИСЦИПЛИНЫ	4
2. МЕСТО ДИСЦИПЛИНЫ В УЧЕБНОМ ПРОЦЕССЕ	4
3. ПЕРЕЧЕНЬ ПЛАНИРУЕМЫХ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ, СООТНЕСЕННЫХ С ПЛАНИРУЕМЫМИ РЕЗУЛЬТАТАМИ ОСВОЕНИЯ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ	5
4. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ	7
4.1 РАСПРЕДЕЛЕНИЕ ТРУДОЁМКОСТИ ДИСЦИПЛИНЫ ПО ВИДАМ РАБОТ ПО СЕМЕСТРАМ	7
4.2 СОДЕРЖАНИЕ ДИСЦИПЛИНЫ	7
4.3 СОДЕРЖАНИЕ ЛЕКЦИЙ/ПРАКТИЧЕСКИХ ЗАНЯТИЙ И КОНТРОЛЬНЫЕ МЕРОПРИЯТИЯ	10
5. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ	12
6. ТЕКУЩИЙ КОНТРОЛЬ УСПЕВАЕМОСТИ И ПРОМЕЖУТОЧНАЯ АТТЕСТАЦИЯ ПО ИТОГАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ	13
6.1. ТИПОВЫЕ КОНТРОЛЬНЫЕ ЗАДАНИЯ ИЛИ ИНЫЕ МАТЕРИАЛЫ, НЕОБХОДИМЫЕ ДЛЯ ОЦЕНКИ ЗНАНИЙ, УМЕНИЙ И НАВЫКОВ И (ИЛИ) ОПЫТА ДЕЯТЕЛЬНОСТИ	13
6.2. ОПИСАНИЕ ПОКАЗАТЕЛЕЙ И КРИТЕРИЕВ КОНТРОЛЯ УСПЕВАЕМОСТИ, ОПИСАНИЕ ШКАЛ ОЦЕНИВАНИЯ	34
7. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ	36
7.1 ОСНОВНАЯ ЛИТЕРАТУРА	36
7.2 ДОПОЛНИТЕЛЬНАЯ ЛИТЕРАТУРА:	37
7.3 НОРМАТИВНЫЕ ПРАВОВЫЕ АКТЫ	37
8. ПЕРЕЧЕНЬ РЕСУРСОВ ИНФОРМАЦИОННО-ТЕЛЕКОММУНИКАЦИОННОЙ СЕТИ «ИНТЕРНЕТ», НЕОБХОДИМЫХ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ	37
9. ПЕРЕЧЕНЬ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ	38
10. ОПИСАНИЕ МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЙ БАЗЫ, НЕОБХОДИМОЙ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ	38
11. МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ СТУДЕНТАМ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ	39
Виды и формы отработки пропущенных занятий.....	39
12. МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПРЕПОДАВАТЕЛЯМ ПО ОРГАНИЗАЦИИ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ	39

Аннотация
рабочей программы учебной дисциплины Б1.Б.12.04 «Базы данных» для
подготовки бакалавра по направлению 09.03.03 «Прикладная информати-
ка»
направленности «Прикладная информатика в экономике»

Цель освоения дисциплины: приобретение студентами теоретических знаний в области концептуального, логического и физического проектирования БД, формирование практических умений и навыков по применению теории при разработке БД в конкретной предметной области.

Место дисциплины в учебном плане: дисциплина включена в обязательную часть учебного плана по направлению подготовки 09.03.03 «Прикладная информатика», дисциплина осваивается в 3-4 семестрах.

Требования к результатам освоения дисциплины: в результате освоения дисциплины формируются следующие компетенции (индикаторы): ОПК-2.1; ОПК-2.2; ОПК-2.3.

Краткое содержание дисциплины:

Дисциплина «Базы данных» посвящена изучению теоретических основ, практических методов и средств построения баз данных, а также вопросов, связанных с жизненным циклом, поддержкой и сопровождением баз данных. Рассматриваются основные понятия баз данных, способы их классификации, принципы организации структур данных и соответствующие им типы систем управления базами данных (СУБД). Изучаются средства и методы хранения данных на физическом уровне. Подробно изучается реляционная модель данных, соответствующие этой модели СУБД, стандартный язык запросов к реляционным СУБД - SQL, методы представления сложных структур данных средствами реляционной СУБД.

Общая трудоемкость дисциплины: 288/8 (часы/зач. ед.).

Промежуточный контроль: зачет в 3 семестре, экзамен в 4 семестре.

1. Цель освоения дисциплины

Целью дисциплины «Базы данных» является освоение теоретических основ проектирования и создания баз данных, приобретение практических навыков в этой области.

Задачи дисциплины:

- освоить теоретические основы реляционных баз данных;
- получить навыки концептуального, логического и физического проектирования БД;
- получить практические навыки работы с такими СУБД как SQL Server.

2. Место дисциплины в учебном процессе

Дисциплина «Базы данных» включена в обязательный перечень дисциплин обязательной части. Дисциплина «Базы данных» реализуется в соответствии с требованиями ФГОС ВО, ОПОП ВО и Учебного плана по направлению 09.03.03 «Прикладная информатика».

Данная дисциплина базируется на знаниях, полученных студентами в ходе изучения курса «Теоретические основы информатики» (1 семестр). Дисциплина «Базы данных» является основополагающей для изучения следующих дисциплин: «Проектирование информационных систем», «Разработка распределенных систем», «Объектно-ориентированное проектирование и программирование».

Рабочая программа дисциплины «Базы данных» для инвалидов и лиц с ограниченными возможностями здоровья разрабатывается индивидуально с учетом особенностей психофизического развития, индивидуальных возможностей и состояния здоровья таких обучающихся.

3. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения образовательной программы

Изучение данной учебной дисциплины направлено на формирование у обучающихся компетенций, представленных в таблице 1.

Таблица 1

Требования к результатам освоения учебной дисциплины

№ п/п	Код компетенции	Содержание компетенции (или её части)	Индикаторы компетенций	В результате изучения учебной дисциплины обучающиеся должны:		
				знать	уметь	владеть
1	ОПК-2	Способен использовать современные информационные технологии и программные средства, в том числе отечественного производства, при решении задач профессиональной деятельности	ОПК-2.1 Знает современные информационные технологии и программные средства, в том числе отечественного производства при решении задач профессиональной	Понятие БД и СУБД, состав СУБД, модели данных, концептуальное, логическое и физическое проектирование, понятие «нормализация», команды языка манипуляции данными	-	-
			ОПК-2.2 Умеет выбирать современные информационные технологии и программные средства, в том числе отечественного производства при решении задач профессиональной деятельности	-	Выбирать приемы и средства проектирования БД для конкретной предметной области.	-
			ОПК-2.3 Владеет навыками применения современных информационных технологий и программных средств, в том числе отечественного производства, при решении задач профессиональной деятельности.	-	-	Навыками проектирования и реализации БД для конкретной предметной области.

4. Структура и содержание дисциплины

4.1 Распределение трудоёмкости дисциплины по видам работ по семестрам

Общая трудоёмкость дисциплины составляет 8 зач. единиц (288 часов), их распределение по видам работ и семестрам представлено в табл. 2.

Таблица 2

Распределение трудоёмкости дисциплины по видам работ по семестрам

Вид учебной работы	Трудоёмкость		
	час.	в т.ч. по семестрам	
		№3	№4
Общая трудоёмкость дисциплины по учебному плану	288	72	216
1. Контактная работа:	120,65	50,25	70,4
Аудиторная работа	120,65	50,25	70,4
<i>в том числе:</i>			
<i>лекции (Л)</i>	50	16	34
<i>практические занятия (ПЗ)</i>	68	34	34
<i>консультации перед экзаменом</i>	2	-	2
<i>контактная работа на промежуточном контроле (КРА)</i>	0,65	0,25	0,4
2. Самостоятельная работа (СРС)	167,35	21,75	145,6
<i>самостоятельное изучение разделов, самоподготовка (проработка и повторение лекционного материала и материала учебников и учебных пособий, подготовка к практическим занятиям, контрольным работам и т.д.)</i>	124,75	12,75	112
<i>Подготовка к зачету</i>	9	9	-
<i>Подготовка к экзамену (контроль)</i>	33,6	-	33,6
Вид промежуточного контроля:	-	зачёт	экзамен

4.2 Содержание дисциплины

Таблица 3

Тематический план учебной дисциплины

Наименование разделов и тем дисциплин (укрупнёно)	Всего	Аудиторная работа			Внеаудиторная работа СР
		Л	ПЗ	ПКР	
Раздел 1. Понятие реляционной модели. Нормализация данных. Проектирование базы данных.	30	9	15	-	6
Раздел 2. Разработка баз данных в SQL Server Management Studio	32,75	7	19	-	6,75
Контактная работа на промежуточном контроле (КРА)	0,25	-	-	0,25	-
Подготовка к зачету	9	-	-	-	9
Итого за 3 семестр	72	16	34	0,25	21,75
Раздел 2. Разработка баз данных в SQL Server Management Studio	108	26	22	-	60
Раздел 3. Использование нескольких таблиц в запросе. Понятие транзакций.	72	8	12	-	52
Контактная работа на промежуточном контроле (КРА)	0,4	-	-	0,4	-

Наименование разделов и тем дисциплин (укрупнёно)	Всего	Аудиторная работа			Внеаудиторная работа СР
		Л	ПЗ	ПКР	
Консультации перед экзаменом	2	-	-	2	-
Подготовка к экзамену (контроль)	33,6	-	-	-	33,6
Итого за 4 семестр	216	34	34	2,4	145,6
Всего	288	50	68	2,65	167,35

Раздел 1. Понятие реляционной модели. Нормализация данных. Проектирование базы данных.

Тема 1. Основные понятия о БД и управлении ими

Общее представление о базах данных и СУБД. Знакомство с основными моделями данных. Сетевая и иерархическая модели данных. Сферы применения реляционной и объектно-ориентированной моделей данных. NoSQL- подход к организации данных.

Тема 2. Архитектура СУБД

Знакомство с общей архитектурой СУБД. Понятие архитектуры «клиент-сервер» и «файл-сервер». Понятие языка определения данных и языка манипулирования данными. Язык запросов SQL. Язык запросов T-SQL (Transact-SQL). Распределенная обработка данных.

Тема 3. Реляционная модель данных

Основные понятия реляционной модели данных. Описание структуры данных, целостности и манипулирования данными. Поля, записи, домены, кортежи и атрибуты. Типы связей. Понятия первичного и внешнего ключа. Простые и составные ключи.

Тема 4. Нормализация данных

Понятие нормализации и ее назначения. Определение нормальных форм от первой до пятой. Описание доменно-ключевой нормальной формы.

Тема 5. Проектирование БД

Основные понятия. Семантическое (концептуальное) моделирование на примере модели «сущность-связь». ER-диаграммы. Понятие логического и физического проектирования. Нотации Чена, UML и другие.

Тема 6. СУБД MS Access: общие сведения

Особенности работы в СУБД MS Access. Компоненты, теоретические аспекты, терминология. Описание типов данных, имеющихся в MS Access. Описание объектов СУБД MS Access: таблиц, форм, запросов, отчетов. Разработка макросов в СУБД MS Access. Разработка базы данных в СУБД MS Access.

Раздел 2. Разработка баз данных в SQL Server Management Studio

Тема 1. Язык SQL. Общие сведения

Предпосылки появления языка SQL. История языка SQL. Общие понятия. Язык определения данных (ЯОД), язык манипулирования данными (ЯМД), язык администрирования БД (ЯАБД). Международные стандарты SQL. Стандарты языка SQL. Основные инструкции. Синтаксис языка SQL. Обзор типов данных. Константы. Выражения. Встроенные функции. Язык Transact-SQL.

Тема 2. Работа со структурой БД в SQL Server Management Studio

Установка и настройка SQL Server. Оболочка SQL Server Management Studio. Создание базы данных. Добавление, обновление и удаление таблиц. Определение ограничений. Работа со структурой, обеспечение целостности данных. Построение диаграмм. Представления в СУБД. Примеры создания БД и работы с таблицами.

Тема 3. Основные операции манипулирования данными

Запросы на выборку: команда SELECT. Запросы на добавление: команда INSERT. Запросы на удаление: команда DELETE. Примеры создания и выполнения запросов.

Тема 4. Дополнительные операции манипулирования данными

Условие WHERE. Сортировка данных при выполнении запроса. Группировка данных запроса. Расчеты в запросе, переименование полей. Обновление данных: команда UPDATE. Слияние данных: команда UNION. Примеры создания и выполнения запросов.

Раздел 3. Использование нескольких таблиц в запросе. Понятие транзакций

Тема 1. Использование нескольких таблиц в запросе

Связывание данных: команда INNER JOIN ON. Принципы связывания таблиц. Команды LEFT JOIN и RIGHT JOIN. Примеры создания и выполнения запросов.

Тема 2. Обработка транзакций

Понятие транзакция. Транзакция и работа в многопользовательском режиме.

4.3 Содержание лекций/практических занятий и контрольные мероприятия

Таблица 4

№ п/п	№ раздела	№ и название лекций/практических занятий	Формируемые компетенции (индикаторы)	Вид контрольного мероприятия	Кол-во часов
1.	Раздел 1. Понятие реляционной модели. Нормализация данных. Проектирование базы данных				
	Тема 1. Основные понятия о БД и управление ими	Лекция №1 Основные понятия о БД и управление ими	ОПК-2.1 ОПК-2.2 ОПК-2.3	-	1
	Тема 2. Архитектура СУБД	Лекция №2 Архитектура СУБД		-	1
	Тема 3. Реляционная модель данных	Лекция №3 Реляционная модель данных		-	2
	Тема 4 Нормализация данных.	Лекция №4 Нормализация данных		-	2
		Практическая работа №1. Нормализация данных		Контрольная работа	2
	Тема 5. Проектирование БД.	Лекция №5 Проектирование БД.		-	1

№ п/п	№ раздела	№ и название лекций/ практических занятий	Формируемые компетенции (индикаторы)	Вид контрольного мероприятия	Кол-во часов
		Практическая работа №2. Концептуальное, логическое и физическое проектирование.		Защита практической работы	4
		Практическая работа №3. Концептуальное, логическое и физическое проектирование. (продолжение)		Контрольная работа	3
	Тема 6. СУБД MS Access: общие сведения	Лекция №6 СУБД MS Access. Объекты MS Access		-	1
		Лекция №7 Разработка баз данных в программе MS Access		-	1
		Практическая работа №4. Разработка баз данных в программе MS Access		Контрольная работа	6
2	Раздел 2. Разработка баз данных в SQL Server Management Studio				
Тема 1. SQL. Общие сведения	Лекция №9 SQL. Общие сведения		-	2	
	Лекция №10 Обзор языка SQL. Синтаксис на примере		-	3	
Тема 2. Работа со структурой БД в SQL Server Management Studio	Лекция №11 Работа со структурой БД в SQL Server Management Studio.		-	3	
	Практическая работа №7. Создание БД в SQL Server Management Studio.		Защита практической работы	4	
Тема 3. Основные операции манипулирования данными	Лекция №12 Оператор SELECT языка SQL		-	4	
	Практическая работа №8. Основы Transact SQL: Простые (однотабличные) выборки данных		Защита практической работы	10	
	Лекция №13 Операторы INSERT и DELETE языка SQL		-	6	
	Практическая работа №9 Основы Transact SQL: Добавление и удаление данных в таблицах		Защита практической работы	12	
Тема 4. Дополнительные операции манипулирования данными	Лекция №14 Итоговые запросы		-	4	
	Лекция №15 Группировка и сортировка в запросах		-	6	
	Лекция №16 Расчеты в запросах		-	5	
	Практическая работа №10 Итоговые запросы		Защита практической работы	6	
	Практическая работа №11 Запросы с группировкой и сортировкой, расчеты в запросах		Тестирование	9	

№ п/п	№ раздела	№ и название лекций/ практических занятий	Формируемые компетенции (индикаторы)	Вид контрольного мероприятия	Кол- во часов
3	Раздел 3. Использование нескольких таблиц в запросе. Понятие транзакций				20
	Тема 1. Использо- вание нескольких таблиц в запросе	Лекция № 17. Сложные за- просы. Команды INNER JOIN ON и ее вариации	ОПК-2.1 ОПК-2.2 ОПК-2.3	-	4
		Практическая работа № 12 Сложные (многотабличные запросы)		Защита практи- ческой работы	7
	Тема 2. Обработка транзакций	Лекция № 18. Обработка транзакций		-	4
Практическая работа № 13 Обработка ошибок. Управле- ние транзакциями. Триггеры		Контрольная работа		5	

Таблица 5

Перечень вопросов для самостоятельного изучения дисциплины

№ п/п	№ раздела и темы	Перечень рассматриваемых вопросов для самостоятельного изучения	Формируемые компетенции (индикаторы)
Раздел 1. Понятие реляционной модели. Нормализация данных. Проектирование базы данных			
1.	Тема 2. Архитектура СУБД	Распределенная обработка данных.	ОПК-2.1 ОПК-2.2 ОПК-2.3
2.	Тема 6. СУБД MS Access: общие сведения	Виды связей в СУБД MS Access. Спо- собы создания запросов. Макросы и программирование.	
Раздел 2. Разработка баз данных в SQL Server Management Studio			
3.	Тема 1. SQL. Общие све- дения	Международные стандарты SQL. Уста- новка и настройка SQL Server.	ОПК-2.1 ОПК-2.2 ОПК-2.3
4.	Тема 3. Работа со струк- турой БД в SQL Server Management Studio		

5. Образовательные технологии

Таблица 6

Применение активных и интерактивных образовательных технологий

№ п/п	Тема и форма занятия		Наименование используемых ак- тивных и интерактивных образо- вательных технологий
1.	Разработка баз данных в программе MS Access	ПЗ	Мастер-класс
2.	Создание БД в SQL Server Manage- ment Studio.	ПЗ	Мастер-класс
3.	Основы Transact SQL: Простые (од- нотабличные) выборки данных	ПЗ	Мастер-класс
4.	Сложные (многотабличные запросы)	ПЗ	Мастер-класс

**6. Текущий контроль успеваемости и промежуточная аттестация по
итогам освоения дисциплины**

**6.1. Типовые контрольные задания или иные материалы, необходимые для
оценки знаний, умений и навыков и (или) опыта деятельности**

1). Типовые задания для проведения контрольных работ (примеры)

Тема Нормализация данных

Вариант 1

Нормализовать БД до 2 нормальной формы

ФИО	Адрес	Должность	Доля ставки	Оклад	Отдел	Паспорт серия
Иванов И.И.	М. ул. Строителей, 12	Менеджер отдела про- даж	1,0	35000	продаж	2203 445675
...						

Вариант 2

Нормализовать БД до 2 нормальной формы

Цена	Вид то- вара	Фирма по- ставщик	Название товара	Количество	Артикул товара	Артикул вида това- ра
200	крупа	ООО «Авант»	пшено	200 кг	0123056	0235641
...						

Вариант 3

Нормализовать БД до 2 нормальной формы

Номер поля	Культура	Площадь поля	Площадь по- сева под куль- турой	Урожайность культуры в посеве	Влажность зерна
01	рожь	150	30	14 ц/га	19
...					

Вариант 4

Нормализовать БД до 2 нормальной формы

ФИО	Город	Код горо- да	Телефонный номер	Вид те- лефона	Время дозвона	Продолжитель- ность звонка
Иванов И.И.	Москва	499	775-36-45	рабочий	17.35	15 мин
...						

Вариант 5

Нормализовать БД до 2 нормальной формы

Номер склада	Товар	Номер полки	Тип полки	Температура хранения	Вместимость полки	Количество полок
01	010233	2	холодная	2-10	20 коробок	6
...						

Тема «Проектирование БД»

В соответствии с вариантом построить ER-диаграммы в нотациях:

- Классическая нотация П. Чена.
- Нотация IDEFIX(Integration Definition for Information Modeling).
- Нотация Дж. Мартина ("вороньи лапки")
- UML (диаграмма классов)

Вариант 1

Прокат автомобилей

Компании, занимающейся прокатом автомобилей, нужно сохранять данные о клиентах, автомобилях, ценах и заказах. Система должна выдавать расписания для автомобилей, подробные данные о доступных автомобилях, отчеты об использовании и конкретного автомобиля, счета для клиентов.

Вариант 2

Школьная библиотека

Школьная библиотека хочет создать систему, в которой будут храниться данные об учениках, справочниках (не выдающихся на дом), книгах, которые можно выдать на дом, и фактах выдачи книг. Система должна выдавать подробные данные о книгах, выданных на данный момент книгах, просроченных книгах, список наличных книг, отчет по выдачам конкретной книги, письма ученикам с напоминанием о просроченных книгах.

Вариант 3

Спортивный клуб

Спортивному клубу нужно хранить данные о членах клуба. Система должна выдавать списки членов клуба, подробные данные об оплативших и не оплативших абонементы, письма о встречах. Система также должна хранить данные о резервировании помещений и оборудования клуба.

Вариант 4

Школьные практики

Школе нужно хранить данные о практиках учеников, в том числе имена учеников, названия работодателей, кто направлен к какому работодателю и любые дополнительные данные. Система должна выдавать списки учеников, отсутствующих в школе в конкретный день из-за практик, списки учеников для работодателей, отчеты об учениках, которым не назначен работодатель, индивидуальные письма работодателям и ученикам.

Вариант 5

Управление складом

Точка быстрого питания хочет создать систему, в которой будут храниться данные о текущих складских запасах, поставках и продажах. Система должна выдавать отчеты о продажах, о текущем состоянии склада и о движении по складу, и генерировать заказы на пополнение склада.

Тема СУБД MS Access: общие сведения

Вариант 1

Магазин музыкальных, компьютерных и видеозаписей ведет для покупателей каталог имеющихся в продаже записей с указанием их розничных цен, жанра, разновидностей жанра, вида носителя записи, имени (названия) исполнителя или автора, названия произведения или записи, года выпуска, производителя и т. д. Кроме того, магазин ведет внутренний учет текущих оптовых цен на записи, количества экземпляров, проданных за отдельные периоды в прошлом, числа еще не распроданных (имеющихся в наличии) экземпляров записей. В реальной ситуации также требуется вести учет заказов на отсутствующие записи, подсчет прибылей/убытков за прошедшие периоды и т. д. Разработать информационную систему МУЗЫКАЛЬНЫЙ (ВИДЕО-) МАГАЗИН.

Вариант 2

Руководитель компьютерной фирмы, выполняющей сборку персональных компьютеров из готовых комплектующих, заказал разработку базы данных, основанной на двух представлениях данных о комплектующих. Одно представление (для клиентов) содержит данные, которые могут отображаться при согласовании с ними комплектности изделия – в ней указаны розничные цены на комплектующие. Цена комплектующих с течением времени может меняться. Второе представление предназначено для внутреннего пользования и анализа результатов деятельности фирмы – в нем содержатся оптовые цены на комплектующие и краткая информация о поставщиках (клиенты предприятия не имеют доступа к данным этой таблицы). Надо обеспечить ведение заказов от клиентов со сроком изготовления и пометкой «оплачено / не оплачено», расчетом суммарной стоимости различных комплектаций персонального компьютера (с указанием розничной стоимости его отдельных комплектующих и т. д.) и всего заказа клиента в целом. Обеспечить оперативный просмотр списка заказов по различным условиям (тип процессора, клиент, стоимость) на любую дату срока изготовления. Разработать информационную систему КОМПЬЮТЕРНАЯ ФИРМА.

Вариант 3

Оптовый склад принимает партии товаров от поставщиков и отпускает его клиентам мелкими партиями. Требуется вести (количественный и/или стоимостный) учет поступающих и отпускаемых товаров, учет поставщиков и клиентов, печатать приходные и расходные накладные. В реальной ситуации также требуется вести бухгалтерские взаиморасчеты склада с поставщиками и клиентами. Разработать информационную систему СКЛАД.

Вариант 4

Агентство по трудоустройству ведет списки лиц, ищущих работу, и списки вакансий. Вакансии поступают от организаций с указанием должности и оклада. В заявках претендентов, кроме анкетных данных, указываются желаемая должность и оклад. Каждая вакансия заполняется несколькими претендентами согласно их анкетным данным, это распечатывается в бумажном виде, а также передается работодателю. Работодатель независимо от агентства отбирает одного из претендентов (или исключает всех), который и должен занять вакансию в базе данных агентства. После этого вакансия и претендент «аннулируются», т. е. они не должны в дальнейшем появляться в списках неудовлетворенных вакансий и претендентов. Разработать информационную систему АГЕНТСТВО ПО ТРУДОУСТРОЙСТВУ.

Вариант 5

Фирма «Окна и Двери» принимает заказы на изготовление пластиковых окон. Каждый заказ содержит, кроме данных о клиенте, данные о размерах окна – высоте и ширине. Окно изготавливают из пластика, погонный метр которого стоит 100 р., и стекла, квадратный метр которого стоит 50 р. Цена с течением времени может меняться. Кроме пластика и стекла, окно может комплектоваться петлями, ручками, подоконниками и т. д. Необходимо обеспечить ведение заказов от клиентов со сроком изготовления и пометкой «оплачено / не оплачено», расчет стоимости окна (себестоимость материалов плюс себестоимость изготовления) и всего заказа в целом. Обеспечить оперативный просмотр списка заказов (размеры окна, клиент, стоимость) на любой вводимый срок изготовления. В реальной ситуации также требуется вести учет расхода и прихода материалов, что усложняет задачу. Разработать информационную систему ПЛАСТИКОВЫЕ ОКНА.

Тема «Обработка транзакций»

Выполнить запросы к базе данных DB_STUDY, в соответствии с вариантом.

Вариант	Задания к выполнению												
1	1	6	11	16	21	26	31	36	41	46	51	56	61
2	2	7	12	17	22	27	32	37	42	47	52	57	62
3	3	8	13	18	23	28	33	38	43	48	53	58	63
4	4	9	14	19	24	29	34	39	44	49	54	59	64
5	5	10	15	20	25	30	35	40	45	50	55	60	65

Сортировка.

1 Вывести все сведения о студентах из таблицы Students и отсортировать результат по коду студента (поле Code_stud).

2 Вывести список фамилий, имен, отчеств преподавателей (поле Name_lector), их должности (поле Post) и ученые степени (поле Science) из таблицы Lectors, отсортировать результат по названиям должностей по возрастанию (использовать поле Post) и по ученым степеням по убыванию (использовать поле Science).

3 Выбрать из таблицы Groups названия групп и номера курсов (поля Name_group, Name_course) и отсортировать результат по полю Name_course по убыванию.

Изменение порядка следования полей.

4 Вывести все сведения о студентах из таблицы Students таким образом, чтобы в результате порядок столбцов был следующим: Code_group, Name, Surname, Lastname, Phone, Birthd

5 Выбрать все поля из таблицы Subjects таким образом, чтобы в результате порядок столбцов был следующим: Name_subject, Code_subject.

2). Типовые задания практических работ (примеры)

Тема «Проектирование БД»

База данных «Кадры предприятия»

Словесное описание предметной области: государственное предприятие. На предприятии существует ряд подразделений.

Каждое подразделение имеет штатное расписание, в котором имеется перечень должностей. Каждая должность имеет название, краткое название, шифр, нижнюю и верхнюю границы разрядов единой тарифной сетки (от 1 до 18). Также известно, сколько единиц каждой должности выделено подразделению. О сотрудниках, работающих на предприятии, необходимо знать всю историю их перемещения – где, в каком подразделении работал сотрудник, на какой должности, какой имел разряд, дату начала и дату окончания работы. Также о сотруднике необходимо хранить личные данные: ФИО, возраст, пол, семейное положение.

Задание:

1. Разработать проект БД в соответствии с предложенным вариантом.
2. Разработать концептуальную, логическую модель базы данных в известных нотациях.

Темы

«Работа со структурой БД в SQL Server Management Studio»,

«Основные операции манипулирования данными»,

«Дополнительные операции манипулирования данными»,

«Использование нескольких таблиц в запросе»

Создание таблиц

Создадим в базе данных Sales пять таблицы. Первая таблица, Customer, будет хранить информацию о клиентах, вторая таблица City - справочник городов, третья, Product, - информацию о товарах, четвертая, Order, будет содержать подробную информацию о заказах и пятая, OrdItem, - о составе заказа (перечне товаров, входящих в заказ). Ниже представлены все поля этих таблиц и их основные свойства.

Имя столбца	Тип данных	Разрешить null	Описание
Customer			
IdCust	int, identity	нет	Уникальный идентификационный номер клиента, на который можно ссылаться в других таблицах
FName	nvarchar(20)	нет	Имя клиента
LName	nvarchar(20)	нет	Фамилия клиента

IdCity	int	нет	Ссылка на номер города
Address	nvarchar(50)	нет	Адрес клиента
Zip	nchar(5)	нет	Почтовый индекс клиента
Phone	phone	да	Телефонный номер клиента
City			
IdCity	int, identity	нет	Уникальный идентификационный номер города
CityName	nvarchar(20)	нет	Название города
Product			
IdProd	int, identity	нет	Уникальный идентификационный номер для каждого товара
Description	nvarchar(100)	нет	Короткое текстовое описание товара
InStock	int	нет	Количество единиц продукта на складе
Order			
IdOrd	int, identity	нет	Уникальный идентификационный номер заказа
IdCust	int	нет	Ссылка на номер клиента
OrdDate	smalldatetime	нет	Дата и время размещения заказа
OrdItem			
IdOrd	int	нет	Ссылка на номер заказа
IdProd	int	нет	Ссылка на номер товара
Qty	int	нет	Количество единиц товара в заказе
Price	money	нет	Цена товара

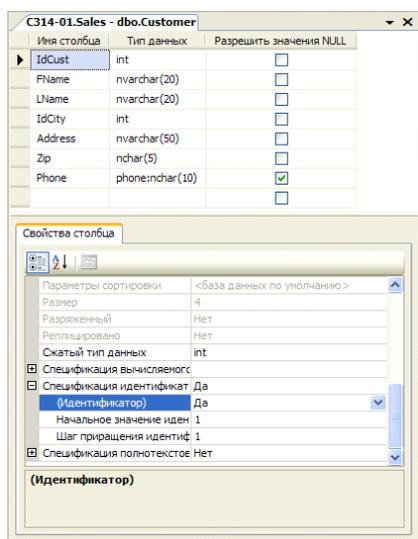
Воспользуемся самым простым, графическим способом для создания таблиц. Сначала создадим таблицу Customer:

В дереве обозревателя объектов в базе данных Sales в контекстном меню узла «Таблицы» выберите команду «Создать таблицу...». В рабочей области должна появиться вкладка с конструктором таблиц.

В первую строку в столбце «Имя столбца» введите IdCust, в столбце «Тип данных» выберите int. Убедитесь, что параметр «Разрешить значения null» отключен.

В нижней половине экрана в разделе «Свойства столбцов» введите описание поля и измените значение параметра «Спецификация идентификатора / (Идентификатор)» на «Да» для того чтобы значения номера клиента формировались автоматически. Свойство «Идентифицирующий столбец» (Identity), обычно используемое совместно с типом данных int, предназначено для автоматического приращения значения на единицу при добавлении каждой новой записи. К примеру, клиент, добавленный в таблицу первым, будет иметь значение идентификатора 1, вторым - 2, третьим - 3, и т.д.

Аналогичным образом введите описания всех остальных полей и закройте окно конструктора таблиц. Введите в качестве имени таблицы Customer. Вновь созданная таблица должна появиться в дереве обозревателя объектов в папке «Таблицы».



Задание для самостоятельной работы: В соответствии с вышеприведенным описанием создайте оставшиеся четыре таблицы: City, Product, Order и OrdItem.

Создание ограничений

Перед тем как начать работать с таблицами следует ограничить вводимые в них данные в целях обеспечения так называемой целостности данных, т.е. ограничить возникновение в базе данных некорректных или противоречивых данных вследствие добавления, изменения или удаления какой-либо записи, например, ввод отрицательной цены или количества товара. Существует четыре типа целостности данных: доменная, сущностная, ссылочная и пользовательская (или бизнес-правила). Рассмотрим основные инструменты, предоставляемые в SQL Server для их реализации.

Обеспечение доменной целостности. Ограничение диапазона данных, вводимых пользователем в поле. Основными инструментами обеспечения доменной целостности являются ограничения проверки и значения по умолчанию.

Использование проверочных ограничений

Ограничения на проверку используются для ограничения данных, принимаемых полем, даже если они имеют корректный тип. Например, поле Zip (почтовый индекс) имеет тип nchar(5), т.е. чисто теоретически оно может принимать буквы. Это может стать проблемой, поскольку не существует почтовых индексов с буквами. Рассмотрим, как создать ограничение на проверку, запрещающее вводить в это поле буквы.

В контекстном меню папки «Ограничения» таблицы Customer выберите команду «Создать ограничение».

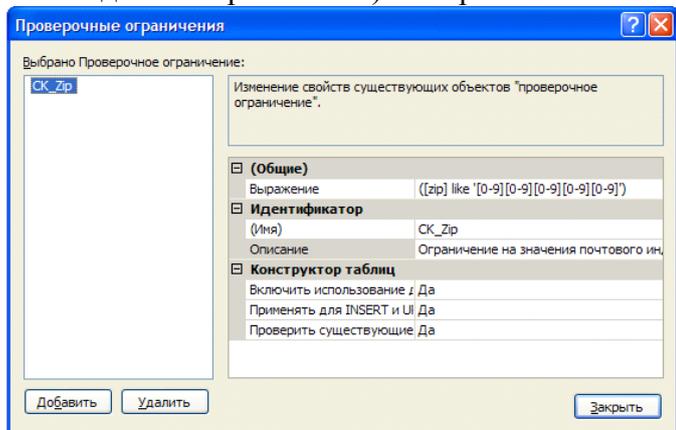
В открывшемся окне «Проверочные ограничения» заполните следующие поля:

Имя: CK_Zip

Выражение: ([zip] like '[0-9][0-9][0-9][0-9][0-9]'). Данное выражение описывает ограничение, принимающее пять символов, которыми могут быть только цифры от 0 до 9.

Описание: Ограничение на значения почтового индекса

Щелкните на кнопке «Закреть» и закройте конструктор таблиц (он был открыт, когда вы начали создавать ограничение) с сохранением изменений.



Задание для самостоятельной работы: Создайте ограничения для полей InStock таблицы Product и Qty, Price таблицы OrdItem, запрещающие ввод в них отрицательных значений. В данном случае выражение проверки будет иметь вид (Имя поля > 0) для полей Qty, Price и (InStock >= 0) для столбца InStock.

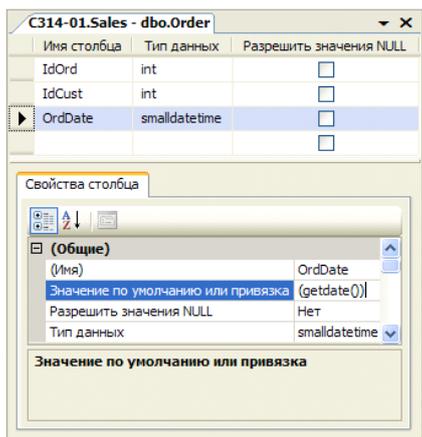
Использование значений по умолчанию

Установка для полей значений по умолчанию это отличный способ избавить пользователя от излишней работы, если значения этих полей во всех записях, как правило, принимают одни и те же значения. Так в таблице заказов Order вполне логично определить по умолчанию значение поля OrdDate (дата заказа) в виде текущей даты. В этом случае при добавлении записи о новом заказе в случае пропуска этого поля оно будет автоматически заполняться значением системной даты. Для создания такого свойства выполните следующие шаги:

Раскройте папку «Столбцы» таблицы Order и в контекстном меню поля «OrdDate» выберите команду «Изменить».

В свойстве столбца «Значение или привязка по умолчанию» введите getdate(). Эта функция T-SQL возвращает текущую системную дату.

Щелкните на кнопке Сохранить и выйдите из конструктора таблиц.



Задание для самостоятельной работы: Установите для поля InStock (количество единиц продукта на складе) таблицы Product в качестве значения по умолчанию ноль.

Обеспечение сущностной целостности. Обеспечение гарантии уникальности записей в таблицах и предотвращение их дублирования. Основными инструментами обеспечения целостности сущностей являются первичные ключи и ограничения уникальности.

Создание первичных ключей

Первичный ключ используется для обеспечения гарантии уникальности каждой записи в таблице. Он состоит из одного (простой ключ) или нескольких (составной ключ) столбцов с гарантированно уникальными значениями для каждой записи таблицы. Если пользователь попытается ввести в поля первичного ключа дублирующее значение будет сгенерирована ошибка и модификация данных будет отменена.

В качестве примера создадим первичный ключ для таблицы Customer. В данном случае идеальным кандидатом на роль первичного ключа выступает столбец IdCust, поскольку значения, содержащиеся в нем, являются уникальными по определению (для него установлено свойство identity). Следует отметить, что в качестве первичного ключа могут быть взяты и реальные атрибуты клиента, например, ИНН, номер страхового свидетельства, серия и номер паспорта вместе взятые (пример составного ключа), но использование различных разновидностей, так называемых, суррогатных ключей (identity, uniqueidentifier) обеспечивает большую степень сущностной целостности (поскольку реальные атрибуты могут все же со временем измениться) и является распространенной практикой. Для создания первичного ключа в таблице Customer выполните следующие шаги:

В контекстном меню таблицы Customer выберите команду «Проект».

В окне конструктора таблиц щелкните правой кнопкой мыши на поле IdCust и выберите команду «Задать первичный ключ» или нажмите кнопку  на панели инструментов. Обратите внимание на то, что слева от поля IdCust теперь отображается значок ключа, указывающий, что поле является первичным ключом.

Закройте конструктор таблиц с сохранением изменений

Имя столбца	Тип данных	Разрешить значения NULL
 IdCust	int	<input type="checkbox"/>
FName	nvarchar(20)	<input type="checkbox"/>
LName	nvarchar(20)	<input type="checkbox"/>
IdCity	int	<input type="checkbox"/>
Address	nvarchar(50)	<input type="checkbox"/>
Zip	nchar(5)	<input type="checkbox"/>
Phone	phone:nchar(10)	<input checked="" type="checkbox"/>

Задание для самостоятельной работы: Аналогичным образом создайте первичные ключи для остальных таблиц в соответствии с ниже приведенной таблицей.

Таблица	Первичный ключ
---------	----------------

City	IdCity
Product	IdProd
Order	IdOrd
OrdItem	IdOrd, IdProd (для выбора нескольких столбцов при установке составного ключа воспользуйтесь клавишами Shift или Ctrl)

Использование ограничений на уникальность

Между ограничениями первичного ключа и ограничениями на уникальность существует два отличия. Первое состоит в том, что первичные ключи используются вместе с внешними ключами для обеспечения целостности ссылок (рассматривается в следующем разделе). Второе отличие заключается в том, что ограничения на уникальность позволяют вставлять в его поля пустые значения (null), чего нельзя делать с первичными ключами. Во всем остальном они служат одной цели - обеспечить уникальность данных, вставляемых в поле. Ограничение на уникальность следует использовать в тех случаях, когда нужно гарантировать, что дублирующие значения не будут добавляться в поле, не являющееся частью первичного ключа, в частности, все потенциальные ключи должны быть организованы в виде ограничений уникальности. Хорошим примером такого поля, требующего ограничение на уникальность, является поле ИНН или серия и номер паспорта, поскольку эти поля должны быть уникальными у каждого человека. Такого идеального кандидата на роль уникального ограничения в нашей таблице Customer нет. Поэтому создадим его по полю Phone, которое также повторяться у разных клиентов не должно.

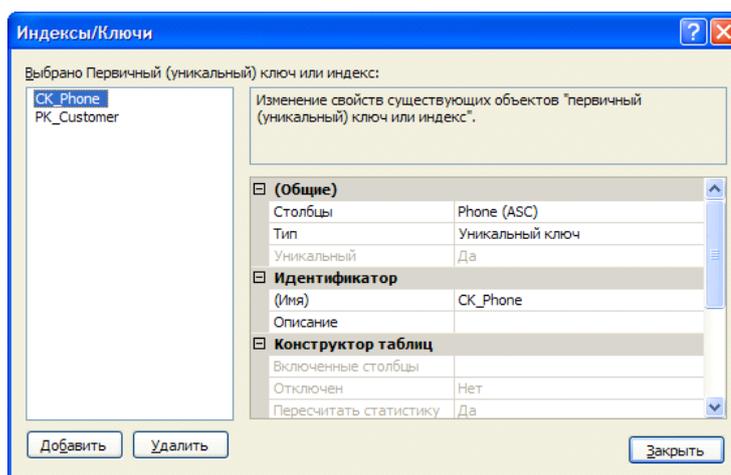
Для открытия конструктора таблиц в контекстном меню таблицы Customer выберите команду «Проект». На панели инструментов нажмите на кнопку «Управление индексами и ключами» .

В открывшемся окне «Индексы и ключи» щелкните кнопку «Добавить» и введите следующие параметры для нового уникального ключа:

Столбцы: Phone

Тип: Уникальный ключ

(Имя): СК_Phone



Закройте конструктор таблиц с сохранением изменений.

Задание для самостоятельной работы: Аналогичным образом создайте ограничение уникальности по полю CityName таблицы City, чтобы обеспечить отсутствие в справочнике городов с одинаковыми названиями, а также по полю Description таблицы Product, чтобы иметь возможность отличить один товар от другого.

Обеспечение целостности ссылок

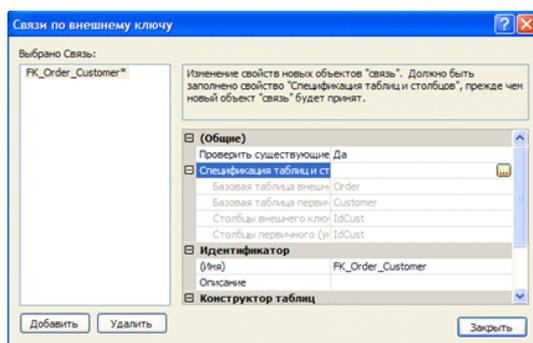
Сейчас в базе данных Sales имеются пять таблиц, которые тесно взаимосвязаны между собой и соответственно данные содержащиеся в них должны быть согласованы и непротиворечивы. Например, в таблице Order не должно быть записей о заказах для клиента, дан-

ные о котором отсутствуют в таблице Customer. Чтобы гарантировать отсутствия в базе данных таких записей необходимо обеспечить целостность ссылок.

Суть обеспечения целостности ссылок очевидна из названия: данные в одной таблице, ссылающиеся на данные из другой таблицы, защищены от некорректного обновления. В терминологии SQL Server это называется декларативной ссылочной целостностью и достигается путем связывания первичного ключа одной из таблиц с внешним ключом другой таблицы (создается так называемое ограничение внешнего ключа).

Внешний ключ используется в комбинации с первичным для связывания двух таблиц по общему столбцу (столбцам). К примеру, можно связать таблицы Customer и Order по столбцу IdCust, который присутствует в обеих таблицах. Поскольку поле IdCust таблицы Customer является его первичным ключом можно использовать поле IdCust таблицы Order в качестве внешнего ключа, который свяжет эти две таблицы. После организации такого ограничения будет невозможно добавить запись в таблицу Order, если в таблице Customer нет записи с соответствующим значением IdCust. Кроме того, при отсутствии каскадирования (рассматривается в следующем разделе) невозможно удалить запись из таблицы Customer при наличии связанных с ней записей в таблице Order, поскольку нельзя оставлять заказ без информации о клиенте. Для создания описанного ограничения внешнего ключа в Management Studio выполните следующие шаги:

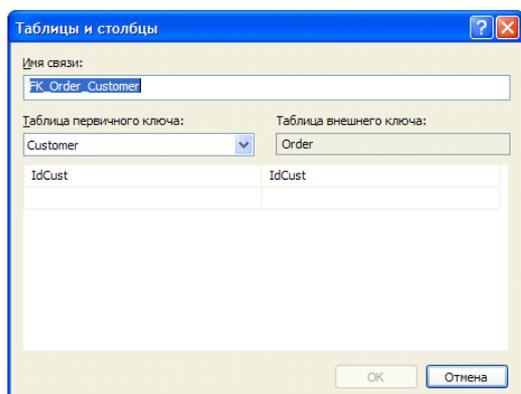
В контекстном меню папки «Ключи» таблицы Order выберите команду «Создать внешний ключ...».



В открывшемся окне «Отношения внешнего ключа» заполните следующие поля:

(Имя): FK_Order_Customer

Спецификация таблиц и столбцов: Для заполнения данного блока щелкните на кнопке с многоточием и в появившемся окне «Таблицы и столбцы» в качестве таблицы первичного ключа выберите Customer, а полей связи - IdCust.



Закройте все открывшиеся окна с сохранением изменений.

Использование каскадной ссылочной целостности

При наличии ограничения внешнего ключа с параметрами по умолчанию вы не можете удалить запись или изменить значение первичного ключа главной таблицы в случае наличия связанных записей в подчиненной таблице (в которой организовано ограничение внешнего ключа). Однако это поведение можно изменить, используя каскадную ссылочную цело-

стность.

Настроить правила каскадирования можно при создании ограничения внешнего ключа в окне «Связи по внешнему ключу» изменяя значения параметров «Правило обновления» и «Правило удаления» блока «Спецификация INSERT и UPDATE».

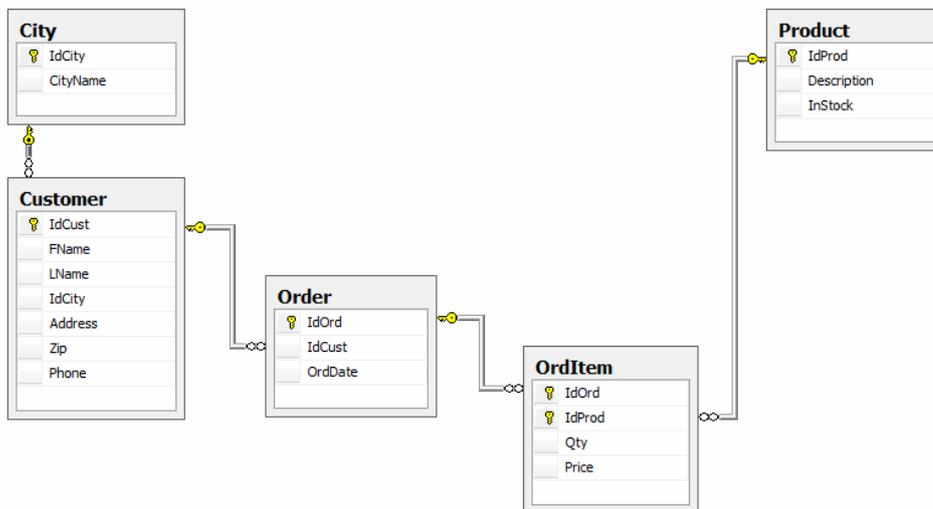
Задание для самостоятельной работы: Создайте ограничение внешнего ключа FK_OrdItem_Order в таблице OrderItem для связи таблиц Order и OrderItem по полю IdOrd. При этом настройте правило каскадного удаления, установив в качестве параметра «Спецификация INSERT и UPDATE\Правило удаления» значение «Каскадно», что приведет к автоматическому удалению всех товаров из заказа при удалении самого заказа.

Использование диаграмм баз данных

Диаграммы базы данных представляют собой графическое отображение схемы (целиком или частично) базы данных с таблицами и столбцами, а также связей между ними. Создадим диаграмму базы данных:

В контекстном меню папки «Диаграммы базы данных» выберите команду «Создать диаграмму базы данных». В диалоговом окне «Добавление таблиц» выберите все таблицы и нажмите на кнопку «Добавить».

Добавив таблицы, щелкните на кнопке «Заккрыть» и вы увидите созданную диаграмму базы данных (на рисунке представлен окончательный вид диаграммы: некоторые связи у вас могут отсутствовать).



Используя диаграмму базы данных ограничения внешнего ключа можно создавать значительно быстрее: лишь перетаскивая поля из одной таблицы в другую. В качестве примера создадим внешний ключ в таблице Customer по полю IdCity для связи с таблицей City:

Выделите в таблице City поле IdCity и, не отпуская кнопку мыши, перетащите его на поле IdCity таблицы Customer.

В диалоговых окнах «Таблицы и столбцы» и «Связь по внешнему ключу» примите настройки по умолчанию.

Сохраните диаграмму базы данных под именем ИЛМ.

Расположите таблицы в канонической форме (главные таблицы выше подчиненных) в соответствии с вышеприведенным рисунком.

Задание для самостоятельной работы: Аналогичным образом создайте связь между таблицами Product и OrdItem по полю IdProduct. Окончательный список связей между таблицами со всеми их характеристиками представлен в следующей таблице:

Главная таблица	Подчиненная таблица	Поле связи (внешний ключ)	Правила каскадирования
City	Customer	IdCity	Без действия
Customer	Order	IdCust	Без действия
Order	OrdItem	IdOrd	Каскадное удаление

Product	OrdItem	IdProd	Без действия
---------	---------	--------	--------------

Задание для самостоятельной работы: После настройки всех ограничений можно заполнить таблицы данными. Для этого в контекстном меню таблицы выберите команду «Изменить первые 200 строк» и появившейся в рабочей области вкладке введите новые записи, заполняя все необходимые столбцы. В процессе внесения данных проверьте работоспособность всех созданных ранее ограничений:

Ограничений проверки: попробуйте ввести в поле Zip (почтовый индекс) таблицы Customer нечисловые значения, а в поля InStock таблицы Product и Qty, Price таблицы OrdItem - отрицательные.

Значений по умолчанию: убедитесь, что при пропуске полей OrdDate и InStock таблиц Order и Product для них устанавливаются значения по умолчанию в виде текущей системной даты и нуля соответственно.

Ограничений первичного и уникального ключа: попробуйте ввести в таблицы записи с дублирующими значениями первичного или уникального ключа.

Ограничений внешнего ключа: попробуйте ввести несогласованные данные в связанные таблицы, например, заказ для несуществующего клиента или удалить запись из любой главной таблицы при наличии связанных записей в подчиненной при отсутствии правил каскадирования.

Правил каскадирования: убедитесь, что при удалении записи из таблицы Order все связанные записи из таблицы OrdItem удаляются автоматически.

Выполнение запросов

Таблица 1. Salespeople (Продавцы)

snum	sname	city	comm
1001	Peel	London	0.12
1002	Serres	San Jose	0.13
1004	Motika	London	0.11
1007	Rifkin	Barcelona	0.15
1003	Alexrod	New York	0.10

snum - уникальный номер, назначенный каждому продавцу ("номер служащего"); sname - имя продавца; city - расположение продавца (город); comm - комиссионные продавцов в десятичной форме.

Таблица 2. Customers (Покупатели)

cnum	cname	city	rating	snum
2001	Hoffman	London	100	1001
2002	Giovanni	Rome	200	1003
2003	Liu	San Jose	200	1002
2004	Grass	Berlin	300	1002
2006	Clemens	London	100	1001
2008	Cisneros	San Jose	300	1007
2007	Pirera	Rome	100	1004

cnum - уникальный номер, назначенный каждому заказчику; cname - имя заказчика; city - расположение заказчика (город); rating - код, указывающий уровень предпочтения данного заказчика перед другими. Более высокий номер указывают на большее предпочтение (рейтинг); snum - номер продавца, назначенного этому заказчику (из таблицы Продавцов)

Таблица 3. Orders (Заказы)

onum	amt	odate	cnum	snum
------	-----	-------	------	------

3001	18.69	10/03/2015	2008	1007
3003	767.19	10/03/2015	2001	1001
3002	1900.10	10/03/2015	2007	1004
3005	5160.45	10/03/2015	2003	1002
3006	1098.16	10/03/2015	2008	1007
3009	1713.23	10/04/2015	2002	1003
3007	75.75	10/04/2015	2004	1002
3008	4723.00	10/05/2015	2006	1001
3010	1309.95	10/06/2015	2004	1002
3011	9891.88	10/06/2015	2006	1001

oNum - уникальный номер данный каждому приобретению; amt - значение суммы приобретений; odate - дата приобретения; cnum - номер заказчика, делающего приобретение (из таблицы Заказчиков); snum - номер продавца, продающего приобретение (из таблицы Продавцов).

1. Напишите команду SELECT, которая бы вывела номер заказа, сумму, и дату для всех строк из таблицы Заказов.

2. Напишите запрос, который вывел бы все строки из таблицы Заказчиков, для которых номер продавца равен 1001.

3. Напишите запрос, который вывел бы таблицу со столбцами в следующем порядке: city, sname, snum, comm.

4. Напишите команду SELECT которая вывела бы оценку (rating), сопровождаемую именем каждого заказчика в San Jose.

5. Напишите запрос, который вывел бы значения snum всех продавцов в текущем порядке из таблицы Заказов без каких бы то ни было повторений

6. Напишите запрос, который отображает все заказы со значениями суммы выше чем \$1,000.

7. Напишите запрос, который находит sname и city для всех продавцов в Лондоне с комиссионными выше 0.1.

8. Напишите запрос к таблице Заказчиков для отображения всех заказчиков с рейтингом не более 100, если они не находятся в Риме.

9. Что будет выведено в результате запроса:

```
SELECT *
```

```
FROM Orders
```

```
WHERE (amt < 1000 OR NOT (odate = '2015-03-10' AND cnum > 2003));
```

10. Что будет выведено в результате запроса:

```
SELECT *
```

```
FROM Orders
```

```
WHERE NOT ((odate = '2015-03-10' OR snum > 1006) AND amt > = 1500);
```

11. Напишите два запроса которые могли бы вывести все заказы за 3 или 4 Октября 2015.

12. Напишите запрос, который выберет всех заказчиков, обслуживаемых продавцами Peel или Motika.

13. Напишите запрос, который может вывести всех заказчиков, чьи имена начинаются с буквы попадающей в диапазон от A до G.

14. Напишите запрос, который выберет всех пользователей чьи имена начинаются с буквы C.

15. Напишите запрос, который выберет все порядки имеющие нулевые значения или NULL в поле `amt` (сумма).
16. Напишите запрос, который сосчитал бы все суммы приобретений на 3 Октября.
17. Напишите запрос, который сосчитал бы число различных не-NULL значений поля `city` в таблице Заказчиков.
18. Напишите запрос, который выбрал бы наименьшую сумму для каждого заказчика.
19. Напишите запрос, который бы выбирал заказчиков в алфавитном порядке, чьи имена начинаются с буквы G.
20. Напишите запрос, который выбрал бы высшую оценку в каждом городе.
21. Напишите запрос, который сосчитал бы число заказчиков, регистрирующих каждый день свои заказы. (Если продавец имел более одного заказа в данный день, он должен учитываться только один раз.)
22. Предположим, что каждый продавец имеет 12% комиссионных. Напишите запрос к таблице заказов, который мог бы вывести номер заказа, номер продавца, и сумму комиссионных продавца для этого заказа.
23. Напишите запрос к таблице Заказчиков который мог бы найти высшую оценку в каждом городе. Вывод должен быть в такой форме: For the city [city], the highest rating is: [rating].
24. Напишите запрос, который выводил бы список заказчиков в нисходящем порядке. Вывод поля оценки (`rating`) должен сопровождаться именем заказчика и его номером.
25. Напишите запрос, который бы вывел список номеров заказов, сопровождающихся именем заказчика, который создавал эти заказы.
26. Напишите запрос, который бы выдавал имена продавца и заказчика для каждого заказа после номера заказа.
27. Напишите запрос, который бы выводил всех заказчиков, обслуживаемых продавцом с комиссионными выше 12%. Выведите имя заказчика, имя продавца и ставку комиссионных продавца.
28. Напишите запрос, который вычислил бы сумму комиссионных продавца для каждого заказа заказчика с оценкой выше 100.
29. Напишите запрос, который бы вывел все пары продавцов, живущих в одном и том же городе. Исключите комбинации продавцов с ними же, а также дубликаты строк, выводимых в обратном порядке
30. Напишите запрос, который вывел бы все пары порядков по данным заказчикам, именам этих заказчиков, и исключал дубликаты из вывода, как в предыдущем вопросе.
31. Напишите запрос, который вывел бы имена (`sname`) и города (`city`) всех заказчиков с такой же оценкой (`rating`) как у Hoffmana. Напишите запрос, использующий поле `snum` Hoffmana, а не его оценку, так чтобы оно могло быть использовано, если его оценка вдруг изменится.
32. Напишите запрос, который бы использовал подзапрос для получения всех заказов для заказчика с именем Cisneros. Предположим, что вы не знаете номера этого заказчика, указываемого в поле `snum`.

33. Напишите запрос, который вывел бы имена и оценки всех заказчиков, которые имеют усредненные заказы.
34. Напишите запрос, который бы выбрал общую сумму всех приобретений в порядках для каждого продавца, у которого эта общая сумма больше, чем сумма наибольшего заказа в таблице.
35. Напишите команду SELECT, использующую подзапрос, которая выберет имена и номера всех заказчиков с максимальными для их городов оценками.
36. Напишите два запроса, которые выберут всех продавцов (по их имени и номеру) которые, в своих городах имеют заказчиков, которых они не обслуживают. Один запрос — с использованием объединения и один — с подзапросом.

3). Типовые тестовых задания

Тема «Дополнительные операции манипулирования данными»

1. Какая SQL команда используется для обновления данных в базе?
 - a. MODIFY
 - b. SAVE
 - c. SAVE AS
 - d. UPDATE
2. Какая SQL команда используется для удаления данных из базы?
 - a. REMOVE
 - b. COLLAPSE
 - c. DELETE
3. Какая SQL команда используется для вставки данных в базу?
 - a. ADD RECORD
 - b. INSERT NEW
 - c. INSERT INTO
 - d. ADD NEW
4. Как выбрать колонку с названием "FirstName" из таблицы "Persons"?
 - a. SELECT Persons.FirstName
 - b. SELECT FirstName FROM Persons
 - c. EXTRACT FirstName FROM Persons
5. Как выбрать все поля из таблицы "Persons"?
 - a. SELECT Persons
 - b. SELECT *.Persons
 - c. SELECT [all] FROM Persons
 - d. SELECT * FROM Persons
6. Как выбрать все записи из таблицы "Persons", где значение поля "FirstName" равно "Peter"?
 - a. SELECT [all] FROM Persons WHERE FirstName LIKE 'Peter'
 - b. SELECT * FROM Persons WHERE FirstName<>'Peter'
 - c. SELECT * FROM Persons WHERE FirstName='Peter'
 - d. SELECT [all] FROM Persons WHERE FirstName='Peter'
7. Как выбрать все записи из таблицы "Persons", где значение поля "FirstName" начинается с буквы "a"?
 - a. SELECT * FROM Persons WHERE FirstName='a'

- b. `SELECT * FROM Persons WHERE FirstName LIKE 'a%'`
 - c. `SELECT * FROM Persons WHERE FirstName LIKE '%a'`
 - d. `SELECT * FROM Persons WHERE FirstName='%a%'`
8. Оператор OR срабатывает, когда ХОТЬ ОДНО условие истинно. AND когда ВСЕ условия истинны.
- a. Ложь
 - b. Истина
9. Как выбрать все записи из таблицы "Persons", где значение поля "FirstName" равно "Peter" и "LastName" равно "Jackson"?
- a. `SELECT FirstName='Peter', LastName='Jackson' FROM Persons`
 - b. `SELECT * FROM Persons WHERE FirstName<>'Peter' AND LastName<>'Jackson'`
 - c. `SELECT * FROM Persons WHERE FirstName='Peter' AND LastName='Jackson'`
10. Как выбрать все записи из таблицы "Persons", где значение поля "LastName" в алфавитном порядке находится между значениями "Hansen" и "Pettersen"?
- a. `SELECT * FROM Persons WHERE LastName>'Hansen' AND LastName<'Pettersen'`
 - b. `SELECT LastName>'Hansen' AND LastName<'Pettersen' FROM Persons`
 - c. `SELECT * FROM Persons WHERE LastName BETWEEN 'Hansen' AND 'Pettersen'`
11. Какое SQL выражение используется для извлечения различных значений?
- a. `SELECT UNIQUE`
 - b. `SELECT DIFFERENT`
 - c. `SELECT DISTINCT`
12. Какая SQL команда используется для упорядочивания результатов?
- a. `SORT BY`
 - b. `ORDER`
 - c. `ORDER BY`
 - d. `SORT`
13. Как выбрать все записи из таблицы "Persons", упорядоченных по полю "FirstName" в обратном порядке?
- a. `SELECT * FROM Persons SORT BY 'FirstName' DESC`
 - b. `SELECT * FROM Persons ORDER BY FirstName DESC`
 - c. `SELECT * FROM Persons SORT 'FirstName' DESC`
 - d. `SELECT * FROM Persons ORDER FirstName DESC`
14. Выберите правильный SQL запрос для вставки новой записи в таблицу "Persons".
- a. `INSERT INTO Persons VALUES ('Jimmy', 'Jackson')`
 - b. `INSERT ('Jimmy', 'Jackson') INTO Persons`
 - c. `INSERT VALUES ('Jimmy', 'Jackson') INTO Persons`
15. Выберите правильный SQL запрос для вставки новой записи в таблицу "Persons", причём в поле "LastName" вставить значение "Olsen".
- a. `INSERT INTO Persons ('Olsen') INTO LastName`
 - b. `INSERT INTO Persons (LastName) VALUES ('Olsen')`
 - c. `INSERT ('Olsen') INTO Persons (LastName)`
16. Как изменить значение "Hansen" на "Nilsen" в колонке "LastName", таблицы Persons?
- a. `UPDATE Persons SET LastName='Nilsen' WHERE LastName='Hansen'`
 - b. `MODIFY Persons SET LastName='Hansen' INTO LastName='Nilsen'`
 - c. `UPDATE Persons SET LastName='Hansen' INTO LastName='Nilsen'`
 - d. `MODIFY Persons SET LastName='Nilsen' WHERE LastName='Hansen'`
17. Как удалить записи, где значение поля "FirstName" равно "Peter"?

- a. DELETE ROW FirstName='Peter' FROM Persons
 - b. DELETE FirstName='Peter' FROM Persons
 - c. DELETE FROM Persons WHERE FirstName = 'Peter'
18. Как вывести количество записей, хранящихся в таблице "Persons"?
- a. SELECT COUNT() FROM Persons
 - b. SELECT COLUMNS() FROM Persons
 - c. SELECT COLUMNS(*) FROM Persons
 - d. SELECT COUNT(*) FROM Persons
19. Какая инструкция языка SQL используется для создания запроса на выборку?
- a. SELECT
 - b. SELECT ... INTO
 - c. INSERT ... INTO
 - d. ORDER BY DELETE
 - e. UPDATE
 - f. TRANSFORM
20. Инструкция языка SQL "CREATE TABLE" используется для:
- a. Создания представления
 - b. Создания индекса
 - c. Обновления таблицы
 - d. Сохранения результатов выборки в новой таблице
 - e. Создания таблицы
21. Какая инструкция языка SQL используется для создания запроса на удаление
- a. SELECT
 - b. SELECT ... INTO
 - c. INSERT ... INTO
 - d. ORDER BY
 - e. DELETE
 - f. UPDATE
 - g. TRANSFORM
22. Инструкция языка SQL "DROP TABLE" используется для:
- a. Создания представления
 - b. Изменения индекса
 - c. Обновления таблицы
 - d. Изменения таблицы
 - e. Создания таблицы
 - f. Удаления таблицы
23. Для группировки данных потребуется предложение:
- a. ORDER BY
 - b. GROUP BY
 - c. HAVING
 - d. Не одно из вышеперечисленных
24. Чтобы исключить дублирующиеся значения из результата выборки, нужно использовать:
- a. SELECT ALL
 - b. SELECT DISTINCT
 - c. SELECT DISTINCT ROW
 - d. SELECT TOP 10% SELECT
 - e. NOT DUPLICATE

25. В таблице Employers есть поле FName, LName и BirthDate. Что произойдёт при выполнении следующей процедуры:
SELECT FName , LName, Birthdate
From Employers
ORDER BY BirthDate DESC, LName;
- Записи отсортируются в порядке убывания дат рождения, а затем по возрастанию фамилий
 - Записи отсортируются в порядке убывания фамилий, а затем по возрастанию дат рождения
 - Записи отсортируются в порядке возрастания фамилий, а затем по убыванию дат рождения
 - Записи отсортируются в порядке возрастания дат рождения, а затем по убыванию фамилий
26. Какие команды можно использовать для того, чтобы упорядочить таблицу заказов (Orders) по дате заказа (поле OrderDate) по возрастанию:
- SELECT ID, Order, OrderDate FROM Orders ORDER BY OrderDate DESC;*
 - SELECT ID, Order, OrderDate FROM Orders ORDER BY 3 ASC;*
 - SELECT * FROM Orders ORDER BY OrderDate;*
 - SELECT ID, Order FROM Orders ORDER BY 2;*
27. Какую итоговую функцию надо выбрать для вычисления количества записей в группе?
- COUNT
 - STDEV
 - AVG
 - VAR
 - SUM
28. В таблице Customers хранятся данные о клиентах, а в таблице Phones – телефоны клиентов. Что выполняет данная инструкция:
SELECT Customers.Lname
FROM Customers INNER JOIN Phones ON Customers.CustomerNum
=Phone.CustomerNum WHERE (Customers.LName Like 'N%')
ORDER BY Customers.LName;
- Выбирает всех клиентов из таблицы Customers, у которых есть телефон
 - Выбирает всех клиентов из таблицы Customers, у которых фамилия начинается на букву N
 - Выбирает всех клиентов из таблицы Customers, у которых фамилия начинается на букву N и при этом есть телефон
 - Выбирает всех клиентов из таблицы Customers, у которых фамилия начинается на букву N и при этом нет телефона
29. Какое условие нужно использовать для отбора связанных данных из двух таблиц?
- INNER JOIN
 - DISTINCTROW
 - LEFT OUTER JOIN
 - CROSS PRODUCT
 - RIGHT OUTER JOIN
30. Какое из условий предложения WHERE позволяет отобрать адреса клиентов из USA и Canada:
- Country = 'USA' AND Country = 'Canada'*
 - Country = 'USA' OR Country = 'Canada'*

- c. Country IN ('USA', 'Canada')
 - d. Country IN ('USA' AND 'Canada') NOT (Country = 'USA' AND Country = 'Canada')
31. Операторы CREATE TABLE и DROP TABLE относятся к группе операторов
- a. управления транзакциями
 - b. манипулирования данными
 - c. определения данных
 - d. определения доступа к данным
32. Операторы INSERT, DELETE, UPDATE относятся к группе операторов
- a. управления курсором
 - b. управления транзакциями
 - c. манипулирования данными
 - d. определения данных
33. Операции COMMIT и ROLLBACK относятся к операциям
- a. запросов
 - b. управления транзакциями
 - c. манипулирования данными
 - d. определения данных
34. Ключевое слово ALL операторе SELECT означает, что
- a. все символы в результирующем наборе заменяются символом *
 - b. в результирующий набор включаются все строки, удовлетворяющие условиям запроса
 - c. в результирующий набор включаются все столбцы из исходных таблиц запроса
35. Операция UPPER(СТРОКА)
- a. преобразует строку к нижнему регистру
 - b. преобразует строку к верхнему регистру
 - c. переводит курсор на одну строку вверх

4). Перечень вопросов, выносимых на промежуточную аттестацию (зачет)

1. Понятие базы данных. Основные элементы БД.
2. 1-ая нормальная форма.
3. Понятие СУБД. Основные элементы СУБД.
4. 2-ая нормальная форма.
5. Основные функции СУБД.
6. 3-я нормальная форма.
7. Модели данных: плоская таблица, сетевая, иерархическая.
8. Нормальная форма Бойса-Кодда.
9. Модели данных: реляционная.
10. 4-ая нормальная форма.
11. Модели данных: объектная.
12. 5-ая нормальная форма.
13. Доменно-ключевая нормальная форма.
14. Интерфейс ADO. Подключение приложения БД к СУБД.
15. Проектирование БД: модель ANSI/SPARC.
16. Ограничения целостности.
17. Проектирование БД: жизненный цикл БД.
18. Ключи: простые, составные, уникальные.
19. Разработка БД и приложений БД.

20. Реализация объектной модели средствами XML.
21. Распределенные и централизованные БД.
22. Объектно-реляционные БД: использование методологии реляционных БД для создания объектной БД.
23. Ограничения целостности при работе с реляционной БД.
24. Обработка данных в БД: клиентский и серверный исполняемый код.
25. Избыточность данных и методы ее устранения при работе с распределенными БД.
26. OLAP и OLTP-технологии.
27. Репликация и синхронизация данных в распределенных реляционных БД.
28. Управление доступом к компонентам БД на основе ACL.
29. Исполнение SQL-запросов на стороне сервера или стороне приложения БД.

**5). Перечень вопросов, выносимых на промежуточную аттестацию
(экзамен)**

1. Способы классификации СУБД (по поддерживаемым типам структур данных; по типам хранимой информации; персональные СУБД и СУБД для коллективного использования; по физической организации: централизованные, распределённые и т.п.).
2. Основные функции СУБД.
3. Модель «сущность-связь». Основные понятия. Область применения.
4. Архитектура СУБД.
5. База данных. Информационные системы. Банк данных. СУБД. Основные понятия и определения.
6. Модели данных. Классификация моделей данных.
7. Иерархическая модель данных. Основные понятия. Область применения. Достоинства и недостатки.
8. Сетевая модель данных. Основные понятия. Область применения. Достоинства и недостатки.
9. Реляционная модель данных. Основные понятия. Область применения. Достоинства и недостатки.
10. Основные понятия реляционной модели данных: отношение, атрибут, домен, кортеж.
11. Целостная часть реляционной модели. Первичный ключ. Свойства ключа. Внешний ключ.
12. Этапы проектирования базы данных. Инфологическое проектирование.
13. Метод моделирования «Сущность - связь» (ER - диаграмма). Примеры.
14. Виды связи между сущностями. Примеры.
15. Логическое проектирование. Правила отображения ER - диаграммы на логическую схему. Примеры.
16. Физическое проектирование.

17. Нормализация таблиц. Проблемы, решаемые при нормализации таблиц.
18. Нормализация таблиц. 1 нормальная форма (1НФ). Примеры.
19. Нормализация таблиц. 2 нормальная форма (2НФ). Функциональная зависимость.
20. Нормализация таблиц. 3 нормальная форма (3НФ). Нормальная форма Бойса-Кодда. (НФБК).
21. Нормализация таблиц. 4 нормальная форма (4НФ). 5 нормальная форма (5НФ).
22. MS Access. Объект Таблица. Назначение. Режимы работы. Свойства поля. Постановка. Связи между таблицами.
30. MS Access. Объект Запросы. Назначение. Режимы работы. Бланк запроса. Условие отбора. Сортировка. Группировка. Запрос с параметром. Вычисляемое поле. Запросы из нескольких таблиц.
31. MS Access. Запросы на выборку. Перекрестный запрос. Запросы на создание, обновление, добавление, удаление таблиц.
32. MS Access. Объект Форма. Назначение. Режимы работы. Подчиненная форма. Элементы управления. Разделы формы.
33. MS Access. Объект Отчеты. Назначение. Режимы работы. Группировка записей. Сортировка. Разделы отчета.
34. История развития языка SQL.
35. Числовые типы данных в MS SQL Server.
36. Символьные типы данных в MS SQL Server.
37. Параметры сортировки в MS SQL Server.
38. Представление дат и времени в MS SQL Server.
39. Многопользовательский доступ к БД в MS SQL Server. Имена входа, пользователи, схемы данных, привилегии.
40. Создание дампа базы данных – назначение, способы.
41. Представления в MS SQL Server. Построение представлений по правилам ANSI SQL, и Transact-SQL.
42. Язык SQL. Выборка данных из таблиц. Оператор SELECT.
43. Язык SQL. Задание условий при выборке данных. Операторы IS NULL, BETWEEN... AND, IN, LIKE, EXISTS, UNIQUE, ALL, ANY.
44. Язык SQL. Упорядочение данных. Группировка данных. Функции агрегирования.
45. Язык SQL. Выборка данных из нескольких таблиц.
46. Язык SQL. Управление объектами базы данных. Создание таблиц базы данных. Ограничения на множество допустимых значений данных.
47. Язык SQL. Управление объектами базы данных. Добавление к таблице новой записи.
48. Язык SQL. Манипулирование данными. Изменение данных, хранящихся в таблице. Удаление данных из таблицы.
49. Предложение INSERT языка SQL. Вставка единственной записи. Вставка множества записей.

50. Структура предложения SELECT, условия выбора (понятие алиасов, возможности параметра WHERE). Соединение таблиц. Выборка вычисляемых значений. Выборка с исключением дубликатов (DISTINCT).

51. Предложение SELECT языка SQL. Сортировка (ORDER BY). Группирование данных (GROUP BY, GROUP BY ... HAVING).

52. Предложение SELECT языка SQL. Встроенные функции.

53. Предложение SELECT языка SQL. Объединение UNION. Квантор существования EXIST и NOT EXIST.

54. Предложение SELECT языка SQL. Выборка с использованием IN, вложенный SELECT. Подзапрос с несколькими уровнями вложенности. Коррелированный подзапрос.

55. Предложение SELECT языка SQL. Соединение таблиц. INNER, LEFT, RIGHT JOIN.

56. Курсоры. DECLARE CURSOR, DROP CURSOR. Операции, требующие использования курсоров.

57. Предложение DELETE языка SQL. Удаление единственной записи. Удаление множества записей. Удаление с подзапросом.

58. Предложение UPDATE языка SQL. Обновление единственной записи. Обновление множества записей. Обновление с подзапросом. Обновление нескольких таблиц.

59. Представления. Определение представления.

60. Транзакции. Свойства транзакций. Способы завершения транзакций.

6.2. Описание показателей и критериев контроля успеваемости, описание шкал оценивания

Для оценки знаний, умений, навыков и формирования компетенции по дисциплине может применяться **балльно-рейтинговая** система контроля и оценки успеваемости студентов.

В основу балльно-рейтинговой системы (БРС) положены принципы, в соответствии с которыми формирование рейтинга студента осуществляется в ходе текущего, промежуточного контроля и промежуточной аттестации знаний.

Таблица 7

Система рейтинговой оценки успеваемости

Баллы	Бальная оценка текущей успеваемости			
За защиту практической работы	2	3	4	5
За контрольную работу	0-14	15-18	19-22	23-25
За тестовые задания	0 – 21	22 – 26	27 – 31	32 – 35
Оценка	Неудовлетворительно	Удовлетворительно	Хорошо	Отлично

Критерии снижения баллов при оценке контрольных работ представлены в таблице 8.

Критерии снижения баллов

Баллы	Описание
-5	Целиком не выполнен пункт задания
-3	Пункт задания выполнен, но с недочетами.

Критерии оценки выполнения практических работ:

Оценка 2 выставляется на защите практической работы в случае, если было выявлено несамостоятельное выполнение задания, то есть студент не ответил на вопросы по заданию.

Оценка 3 выставляется на защите практической работы в случае, если задание имеет крупные недостатки в реализации: недружелюбный интерфейс, неточность реализации функционала, отсутствие венгерской нотации, защиты ввода.

Оценка 4 выставляется на защите практической работы в случае, если задание имеет незначительные недостатки в реализации: отсутствие венгерской нотации, защиты ввода.

Оценка 5 выставляется на защите практической работы в случае, если задание не имеет недостатков в реализации.

Критерии оценки тестовых заданий:

За каждый правильный ответ начисляется 1 балл рейтинга, общая сумма баллов - 35. Ответ, в котором отмечены не все правильные значения или отмечены лишние значения, считается неверным, и балл за него не начисляется.

Студенты, получившие за контрольное мероприятие оценку «неудовлетворительно», обязаны пройти его повторно и получить минимально допустимое количество баллов.

Итоговая сумма баллов

Виды текущего контроля	Количество видов текущего контроля	Количество баллов за единицу	Количество баллов
Защита практических работ	8	5	40
Контрольная работа	4	25	100
Тестовые задания	1	35	35
Всего	-	-	175

Балльно-рейтинговая система контроля успеваемости

Шкала оценивания	Экзамен	Зачет
140-175	Отлично	зачтено
111-139	Хорошо	
84-110	Удовлетворительно	
0-83	Неудовлетворительно	незачтено

Студенты, набравшие по итогам балльно-рейтинговой системы в 3 семестре более 83 баллов, могут претендовать на получение оценки «зачтено», а по итогам 4 семестра на получение экзаменационной оценки, соответствующей набранным баллам рейтинга в таблице 10.

Экзамен по дисциплине проводится в традиционной форме устного ответа на вопросы билета и выполнения практической задачи на ПК и не входит в балльно-рейтинговую систему оценки. Описание шкалы оценивания экзамена приведено в оценочных материалах дисциплины. Студенты, не выполнившие ни одной предусмотренной работы (или же сдавшие эти работы на неудовлетворительные оценки), до экзамена не допускаются.

7. Учебно-методическое и информационное обеспечение дисциплины

7.1 Основная литература

1. Волк, В. К. Базы данных. Проектирование, программирование, управление и администрирование : учебник / В. К. Волк. — Санкт-Петербург : Лань, 2020. — 244 с. — ISBN 978-5-8114-4189-1. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/126933> (дата обращения: 24.08.2020). — Режим доступа: для авториз. пользователей.

2. Орлова, И. В. Информатика. Практические задания : учебное пособие / И. В. Орлова. — Санкт-Петербург : Лань, 2019. — 140 с. — ISBN 978-5-8114-3608-8. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/113400> (дата обращения: 24.08.2020). — Режим доступа: для авториз. пользователей.

3. Белугина, С. В. Архитектура компьютерных систем. Курс лекций : учебное пособие / С. В. Белугина. — Санкт-Петербург : Лань, 2020. — 160 с. — ISBN 978-5-8114-4489-2. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/133919> (дата обращения: 24.08.2020). — Режим доступа: для авториз. пользователей.

4. Тенгайкин, Е. А. Организация сетевого администрирования. Сетевые операционные системы, серверы, службы и протоколы. Практические работы : учебное пособие / Е. А. Тенгайкин. — Санкт-Петербург : Лань, 2020. — 100 с. — ISBN 978-5-8114-4763-3. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/139326> (дата обращения: 24.08.2020). — Режим доступа: для авториз. пользователей.

7.2 Дополнительная литература:

1. Соснин, П. И. Архитектурное моделирование автоматизированных систем : учебник / П. И. Соснин. — Санкт-Петербург : Лань, 2020. — 180 с. — ISBN 978-5-8114-3919-5. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/130183> (дата обращения: 24.08.2020). — Режим доступа: для авториз. пользователей.
2. Тенгайкин, Е. А. Организация сетевого администрирования. Сетевые операционные системы, серверы, службы и протоколы. Лабораторные работы : учебное пособие / Е. А. Тенгайкин. — Санкт-Петербург : Лань, 2020. — 128 с. — ISBN 978-5-8114-4734-3. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/136178> (дата обращения: 24.08.2020). — Режим доступа: для авториз. пользователей.
3. Никифоров, С. Н. Методы защиты информации. Защищенные сети : учебное пособие / С. Н. Никифоров. — Санкт-Петербург : Лань, 2018. — 96 с. — ISBN 978-5-8114-3099-4. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/110935> (дата обращения: 24.08.2020). — Режим доступа: для авториз. пользователей.
4. Петренко, В. И. Защита персональных данных в информационных системах. Практикум : учебное пособие / В. И. Петренко, И. В. Мандрица. — Санкт-Петербург : Лань, 2019. — 108 с. — ISBN 978-5-8114-3311-7. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/111916> (дата обращения: 24.08.2020). — Режим доступа: для авториз. пользователей.

7.3 Нормативные правовые акты

1. Федеральный закон «Об информации, информационных технологиях и о защите информации» от 06.04.2011 N 65-ФЗ.
2. ГОСТ 34.321-96. Информационные технологии. Система стандартов по базам данных. Эталонная модель управления данными"

8. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины

Для самостоятельного изучения разделов и подготовки к занятиям могут быть использованы следующие ресурсы:

1. <https://habr.com/ru> – русскоязычный веб-сайт в формате коллективного блога с элементами новостного сайта, созданный для публикации новостей, аналитических статей, мыслей, связанных с информационными технологиями, бизнесом и интернетом (открытый доступ)
2. <http://www.sql-ex.ru> – русскоязычный веб-ресурс для приобретения навыков в написании операторов манипуляции данными языка SQL. (Открытый доступ)

9. Перечень программного обеспечения

В процессе изучения дисциплины рекомендуется использовать следующее программное обеспечение: ОС Windows, Microsoft Office, NetOp School (опционально).

Таблица 11

Перечень программного обеспечения

№ п/п	Наименование раздела учебной дисциплины	Наименование программы	Тип программы	Автор	Год разработки
1	Раздел 1	MS Access	обучающая	Разработчик фирма Microsoft	2007
2	Разделы 2-3	Microsoft SQL Server 2012/2016 SQL Server Management Studio	обучающая	Разработчик фирма Microsoft	2012-2016

10. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине

Освоение теоретических основ курса «Базы данных» предусматривает прослушивание и проработку материалов лекций, работу с рекомендованными литературными источниками и интернет-ресурсами. Лекции читаются в аудиториях, оснащенных мультимедийной техникой, на основе подготовленных лектором презентаций с применением активных и интерактивных образовательных технологий.

Практические навыки по курсу «Базы данных» приобретаются путем выполнения основных работ и самостоятельной работы.

Для проведения практических занятий по дисциплине «Базы данных» необходим компьютерный класс с предустановленным на ПЭВМ программным обеспечением, указанным в п. 9.

Таблица 12

Сведения об обеспеченности специализированными аудиториями, кабинетами, лабораториями

Наименование специальных помещений и помещений для самостоятельной работы (№ учебного корпуса, № аудитории)	Оснащенность специальных помещений и помещений для самостоятельной работы
Аудитория для проведения занятий лекционного типа, <i>групповых и индивидуальных консультаций</i> . № 108, уч. корпус № 2	Видеопроектор 3500 Лм
Аудитория для проведения практических занятий, <i>курсового проектирования, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации</i> . №УИТ-07, уч. корпус №12	Персональные компьютеры в количестве 26 штук
Аудитория для проведения практических	Персональные компьютеры в количестве 22 штук

Наименование специальных помещений и помещений для самостоятельной работы (№ учебного корпуса, № аудитории)	Оснащенность специальных помещений и помещений для самостоятельной работы
ских занятий, <i>курсового проектирования, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации. №УИТ-12, уч. корпус №12</i>	
Центральная научная библиотека имени Н.И. Железнова,	Читальные залы библиотеки
Общежития РГАУ-МСХА им. К.А. Тимирязева	Комнаты для самоподготовки

11. Методические рекомендации студентам по освоению дисциплины

Посещение лекционных (с конспектированием рассматриваемых вопросов) и практических занятий (с выполнением практических работ), а также проработка рекомендуемой литературы являются необходимым и достаточным условием для получения знаний, практических умений и навыков по изучаемой дисциплине.

Подготовка студентов к занятиям носит индивидуальный характер, но такая подготовка должна включать чтение конспектов лекций и рекомендуемой литературы, что позволяет усвоить необходимые знания по изучаемой теме. Для получения консультаций по вопросам, ответы на которые студент не смог найти в процессе проработки материалов, предусмотрено внеаудиторное время.

Самостоятельная работа студентов организуется в объеме, предусмотренном данной рабочей программой. Самостоятельная работа формирует навыки поиска необходимой информации и способствует лучшему усвоению материала.

Виды и формы отработки пропущенных занятий

Студент, пропустивший лекционное занятие, обязан представить конспект пропущенной лекции. При пропуске практического занятия студент обязан получить у преподавателя индивидуальный вариант, выполнить и защитить его. Прием и защита индивидуального задания проводится в часы и дни, устанавливаемые преподавателем.

12. Методические рекомендации преподавателям по организации обучения по дисциплине

Реализация компетентностного подхода должна предусматривать широкое использование в учебном процессе активных и интерактивных форм проведения занятий (работы с ПК, мастер-классов и прочее) в сочетании с внеаудиторной работой с целью формирования и развития профессиональных навыков обучающихся.

Контроль знаний студентов проводится в формах текущей и промежуточной аттестаций. Текущая аттестация студентов проводится постоянно на прак-

тических занятиях с помощью контроля результатов выполнения практических заданий, контрольных работ, а также на контрольной неделе. Промежуточная аттестация студентов проводится в форме зачета (3 семестр), экзамена (4 семестр).