

	Факультет	Математики, физики и информатики
	Кафедра	Информатики и информационных технологий
	Направление подготовки	02.03.02 Фундаментальная информатика и информационные технологии
	Профиль	Открытые информационные системы
		Технологии баз данных

Министерство образования и науки Российской Федерации
 Федеральное государственное бюджетное образовательное учреждение
 высшего образования
 «Тульский государственный педагогический университет им. Л. Н. Толстого»
 ФГБОУ ВО «ТГПУ им.Л.Н.Толстого»

УТВЕРЖДЕНА

на заседании Ученого совета университета

Протокол № 2

«11» февраля 2016 г.

Учебная программа дисциплины «Технологии баз данных»

Трудоемкость: 7 зачетных единиц (3 в 5 семестре и 4 в 6 семестре)

Квалификация (степень) выпускника: бакалавр

Форма обучения: очная

Рассмотрена на заседании кафедры информатики и информационных технологий
 протокол протокол № 3 от 18 ноября 2015 г.

Заведующий кафедрой  А.В.Якушин

Одобрена на заседании Ученого совета факультета
 математики, физики и информатики
 протокол № 5 от «17» декабря 2015 г.

Декан  И.Ю. Реброва

СОДЕРЖАНИЕ

1. ПЕРЕЧЕНЬ ПЛАНИРУЕМЫХ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ, СООТНЕСЕННЫХ С ПЛАНИРУЕМЫМИ РЕЗУЛЬТАТАМИ ОСВОЕНИЯ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ.....	3
2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП.....	3
3. ОБЪЕМ ДИСЦИПЛИНЫ И ВИДЫ УЧЕБНОЙ РАБОТЫ.....	4
4. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ, СТРУКТУРИРОВАННОЕ ПО ТЕМАМ, С УКАЗАНИЕМ ОТВЕДЕННОГО НА НИХ КОЛИЧЕСТВА АКАДЕМИЧЕСКИХ ЧАСОВ И ВИДОВ УЧЕБНЫХ ЗАНЯТИЙ.....	5
5. ПЕРЕЧЕНЬ УЧЕБНО-МЕТОДИЧЕСКОГО ОБЕСПЕЧЕНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ «ТЕХНОЛОГИИ БАЗ ДАННЫХ».....	8
6. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ	8
6.1. Перечень компетенций с указанием этапов их формирования в процессе освоения образовательной программы.....	8
6.2. Описание показателей, критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания	9
6.3. Типовые контрольные задания и иные материалы, характеризующие этапы формирования компетенций в процессе освоения образовательной программы.....	10
6.4. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и/или опыта деятельности, характеризующие этапы формирования компетенций	25
7. ПЕРЕЧЕНЬ ОСНОВНОЙ И ДОПОЛНИТЕЛЬНОЙ УЧЕБНОЙ ЛИТЕРАТУРЫ, НЕОБХОДИМОЙ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ	27
7.1 Основная литература:	27
7.2 Дополнительная литература:.....	27
8. ПЕРЕЧЕНЬ РЕСУРСОВ ИНФОРМАЦИОННО-ТЕЛЕКОММУНИКАЦИОННОЙ СЕТИ «ИНТЕРНЕТ», НЕОБХОДИМЫХ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ	28
9. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ .	29
10. ПЕРЕЧЕНЬ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, ИСПОЛЬЗУЕМЫХ ПРИ ОСУЩЕСТВЛЕНИИ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ, ВКЛЮЧАЯ ПЕРЕЧЕНЬ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ И ИНФОРМАЦИОННЫХ СПРАВОЧНЫХ СИСТЕМ.....	29
11. ОПИСАНИЕ МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЙ БАЗЫ, НЕОБХОДИМОЙ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ «ТЕХНОЛОГИИ БАЗ ДАННЫХ».....	29
12. АННОТАЦИЯ РАБОЧЕЙ ПРОГРАММЫ ДИСЦИПЛИНЫ «ТЕХНОЛОГИИ БАЗ ДАННЫХ	30
13. ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ К РАБОЧЕЙ ПРОГРАММЕ ДИСЦИПЛИНЫ «ТЕХНОЛОГИИ БАЗ ДАННЫХ»	31

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

Достижение планируемых результатов обучения, соотнесенных с общими целями и задачами ОПОП, является целью освоения дисциплины (модуля).

Планируемые результаты освоения образовательной программы (код и название компетенции)	Планируемые результаты обучения	Этапы формирования компетенции в процессе освоения образовательной программы
Способность использовать современные инструментальные и вычислительные средства (ПК-3)	<p>Выпускник знает:</p> <ul style="list-style-type: none"> • основные понятия в области теории баз данных; • основные технологии обработки баз данных; • этапы жизненного цикла баз данных; • виды и способы организации запросов к данным в реляционных моделях; • способы программирования и администрирования баз данных. <p>Умеет:</p> <ul style="list-style-type: none"> • выбирать технологию для проектирования реляционной модели данных; • разрабатывать структуру модели данных; • реализовывать этапы жизненного цикла баз данных; • использовать технологии баз данных для обработки табличных данных; • представлять отчеты по результатам обработки данных средствами СУБД. <p>Владеет и (или) имеет опыт деятельности</p> <ul style="list-style-type: none"> • основными методами, способами и средствами получения, хранения, обработки информации в реляционных моделях данных; • типовыми технологиями разработки баз данных 	1,2 этап из 3 (5,6 семестр)

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП

Дисциплина «Технологии баз данных» относится к дисциплинам вариативной части обязательного профессионального цикла дисциплин направления. Изучение данной дисциплины базируется на освоении студентами дисциплин базовой части «Информатика и информационные технологии», «Вводный курс программирования», «Языки программирования», вариативной части профессионального цикла «Основы программирования».

К началу изучения дисциплины студенты должны владеть знаниями в области представления и моделирования данных, видов и назначения информационных технологий, умениями обрабатывать табличную информацию средствами специализированных информационных технологий и составлять программные коды на языках программирования объектно-ориентированной парадигмы.

Освоение данной дисциплины необходимо для качественного изучения дисциплин ООП, базирующихся на разработке и использовании технологии разработки баз данных, для выработки умений поиска, анализа и обработки данных в реляционных моделях, для квалифицированного использования баз данных в профессиональной деятельности, для реализации программной разработки, которой посвящена выпускная квалификационная работа.

3. ОБЪЕМ ДИСЦИПЛИНЫ И ВИДЫ УЧЕБНОЙ РАБОТЫ

Вид учебной работы	Объем часов/ зачетных единиц по формам обучения
	очная
Максимальная учебная нагрузка (всего)	252/7 108/3 (5 семестр) 144/3 (6 семестр)
Контактная работа обучающихся с преподавателем (всего)	88(44+44)
в том числе:	
лекции с применением мультимедийных технологий и раздаточным материалом для студентов	32 (16+16)
лабораторные занятия с использованием современных информационных технологий по разработке алгоритмов и программ	52 (26+26)
контрольные работы	4 (2+2)
Самостоятельная работа студента (всего)	128 (64+64)
в том числе:	
внеаудиторная самостоятельная работа при подготовке к лабораторным занятиям	80 (30+50)
подготовка к контрольной работе	8 (4+4)
подготовка учебного проекта	20 (10+10)
Выполнение заданий для самостоятельной работы в модульной объектно-ориентированной динамической учебной среде Moodle	20 (10+10)
<i>Промежуточная аттестация (5 семестр) в форме: зачета</i>	
<i>Промежуточная аттестация (6 семестр) в форме: экзамена</i>	36

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

Наименование тем (разделов).	Количество академических или астрономических часов по видам учебных занятий			
	Занятия лекционного типа	Занятия лабораторного типа	Другие виды учебных занятий	Самостоятельная работа обучающихся
Раздел 1. Теория баз данных	8	12		28
Тема 1.1. Введение в предмет. Основные понятия. Модели данных	2	4		8
Тема 1.2. Представление данных в информационных системах	2	4		8
Тема 1.3. Реляционная модель данных	4	4		12
Раздел 2. Проектирование баз данных	8	14		28
Тема 2.1. Этапы проектирования баз данных	2	2		7
Тема 2.2. Метод нормальных форм	2	4		7
Тема 2.3. Метод ER-диаграмм	2	2		7
Тема 2.4. Инструментальные средства проектирования структур данных	2	6		7
Контроль самостоятельной работы студентов			2	
Подготовка к зачету				8
ИТОГО за 5 семестр	16	26	2	64
Раздел 3. Язык запросов SQL	8	6		32
Тема 3.1. Операторы определения данных	2	2		10
Тема 3.2. Оператор выборки данных	2	2		10
Тема 3.3. Оператор манипулирования данными	4	2		12
Раздел 4. Программирование и администрирование баз данных	8	20		32
Тема 4.1. Программирование и управление транзакциями	2	4		6
Тема 4.2. Защита баз данных; целостность и сохранность баз данных.	2	6		8
Тема 4.3. Современные тенденции построения баз данных. Объектно-реляционный подход к моделированию систем	2	6		8
Тема 4.4. Промышленные системы управления базами данных	2	4		10
Контроль самостоятельной работы студентов			2	
Подготовка к экзамену				36
ИТОГО за 6 семестр	16	26	2	100
ИТОГО	32	52	4	164

Раздел 1. Теория баз данных**Тема 1.1. Введение в предмет. Основные понятия. Модели данных**

Сферы применения баз данных. История развития автоматизированных систем обработки данных. Суть концепции баз данных. Перспективы развития баз данных. База данных, СУБД, банк данных. Предметная область. Объекты и атрибуты, связи между объектами и атрибутами объектов. Модель предметной области. Концептуальная модель. Типы моделей данных. Реляционная модель данных. Основные определения: отношения, домены, кортежи, атрибуты. Схема отношения, его степень и мощность. Реляционная БД. Первичный ключ. Свойства отношений реляционной БД. Ограничения на отношения, основные операции над отношениями.

Лабораторная работа 1. Введение в технологии баз данных и СУБД

Лабораторная работа 2. Реляционная модель данных

Тема 1.2. Представление данных в информационных системах

Множества и отношения. Основные операции над множествами. Три уровня представления данных в автоматизированных информационных системах. Логический уровень, уровень хранения, физический уровень. Понятия логической и физической независимости данных. Понятие схемы и подсхемы. Модели данных. Основные операции над данными. Понятие целостности данных. Ограничения целостности. Основные виды связей между различными типами объектов. Выбор модели данных

Лабораторная работа 3. Логическая модель данных

Лабораторная работа 4. Физическая модель данных

Тема 1.3. Реляционная модель данных

Структуры данных реляционной модели. Использование отношений для представления данных. Ограничения модели. Функциональные зависимости, правила вывода функциональных зависимостей, полная функциональная зависимость. Ключи отношений.

Лабораторная работа 5. Структуры данных реляционной модели. Ограничения модели

Лабораторная работа 6. Функциональные зависимости, правила вывода функциональных зависимостей, полная функциональная зависимость

Раздел 2. Проектирование баз данных**Тема 2.1. Этапы проектирования баз данных**

Цели и задачи проектирования БД, жизненный цикл приложения БД. Этапы проектирования: определение требований к системе, анализ предметной области, работа с документами, представление результатов анализа, разработка реализаций, сравнение вариантов и выбор наилучшего, конструирование, эксплуатация и сопровождение БД

Лабораторная работа 7. Цели и задачи проектирования БД, жизненный цикл приложения БД

Тема 2.2. Метод нормальных форм

Концепция нормальных форм, аномалии удаления, добавления и изменения. Процесс нормализации отношения, способы нормализации. Взаимосвязи между атрибутами. Нормальные формы схем отношений. Первая нормальная форма. Вторая нормальная форма. Третья нормальная форма. Нормальная форма Бойсса-Кодда. Многозначные зависимости. Аксиомы многозначных зависимостей. Четвертая нормальная форма. Пятая нормальная форма.

Лабораторная работа 8. Нормальные формы отношений: низкая степень нормализации

Лабораторная работа 9. Нормальные формы отношений: высокая степень нормализации

Тема 2.3. Метод ER-диаграмм

Концепция ER-модели, основные понятия ER-технологии. Тип сущности, тип связи, атрибуты, ключи, структурные ограничения: степень связи, класс принадлежности. Графические элементы диаграмм сущностей – связей в различных нотациях: ER-диаграмма, IEEFX1 стандарт. Этапы проектирования и правила формирования отношений.

Лабораторная работа 10. Концепция ER-модели, основные понятия ER-технологии. Стандарты в области проектирования.

Тема 2.4. Инструментальные средства проектирования структур данных

Case-средства: назначение, основные возможности. Жизненный цикл информационной системы. BPwin. Принципы построения модели IDEF0: контекстная диаграмма, субъект моделирования, цель и точка зрения. Диаграммы IDEF0: контекстная диаграмма, диаграммы декомпозиции, диаграммы дерева узлов, диаграммы только для экспозиции (FEO), Работы (Activity). Стрелки (Arrow). Туннелирование стрелок. Нумерация работ и диаграмм. Каркас диаграммы. Слияние и расщепление моделей. Создание отчетов. Стоимостный анализ: объект затрат, двигатель затрат, центр затрат. Свойства, определяемые пользователем (UDP). Диаграммы потоков данных (Data Flow Diagramming): работы, внешние сущности (ссылки), потоки работ, хранилища данных. Метод описания процессов IDEF3: работы, связи, объекты ссылок, перекрестки. Имитационное моделирование: источники и стоки, очереди, процессы.

Лабораторная работа 11. Case-средства: назначение, основные возможности. Жизненный цикл информационной системы

Лабораторная работа 12. Диаграммы

Лабораторная работа 13. Имитационное моделирование: источники и стоки, очереди, процессы

Раздел 3. Язык запросов SQL

Тема 3.1. Операторы определения данных. Операторы описания структуры данных (DDL), создание и удаление основных объектов базы данных: таблиц, индексов, представлений (view). Ограничения: назначения, виды (primary key, foreign key, unique, not null, check, default), синтаксис. Поддержка целостности данных: внешние и родительские ключи.

Лабораторная работа 1. Операторы описания структуры данных: DDL-запросы

Тема 3.2. Оператор выборки данных. Оператор выборки записей. Агрегатные функции. Группировка и упорядочение записей. Применение выражений и функций в операторе SELECT, классификация функций, агрегация данных, условные выражения (>, <, !=, like, between, in, not, is null)

Лабораторная работа 2. Оператор выборки записей: выражения и функции в операторе SELECT

Тема 3.3. Оператор манипулирования данными. Полный и сокращенный синтаксис оператора добавления данных. Синтаксис операторов на обновление и удаление данных

Лабораторная работа 3. DML-запросы: операторы добавления, обновления и удаления данных

Раздел 4. Программирование и администрирование

Тема 4.1. Программирование и управление транзакциями. Модель транзакции. Свойства транзакции. Журнализация. Проблемы многопользовательских систем. Блокировка. Алгоритмы блокировки. Целостность и восстановление баз данных. Управление обменом с внешней памятью, дисциплины обслуживания обращений к внешним ЗУ. Назначение хранимых процедур и их использование. Свойства ACID транзакций.

Лабораторная работа 4. Транзакции. Целостность и восстановление баз данных

Лабораторная работа 5. Хранимые процедуры и триггеры

Тема 4.2. Защита баз данных; целостность и сохранность баз данных. Создание и удаление баз данных. Защита баз данных. Управление учетными записями и правами доступа. Резервное копирование и восстановление баз данных. Контролируемая избыточность данных. Обеспечение защиты данных в банках данных. Обеспечение целостности и достоверности данных. Целостность и сохранность баз данных.

Лабораторная работа 6. Создание и удаление баз данных

Лабораторная работа 7. Управление учетными записями и правами доступа. Резервное копирование и восстановление баз данных

Лабораторная работа 8. Обеспечение целостности и достоверности данных

Тема 4.3. Современные тенденции построения баз данных. Объектно-реляционный подход к моделированию систем. Проблема адекватности отображения предметных областей в модели данных. Проблема избыточности и независимости данных. Проблема статичности реляционной модели данных. Реорганизация модели данных и идентификация объектов в базе данных. Объектно-реляционный подход к проектированию моделей баз данных. Предпосылки и достоинства подхода. Средства идентификации объектов в модели и средства моделирования. Этапы проектирования объектно-реляционной модели. Запланированные и незапланированные запросы. Оптимизация запросов в базе данных.

Лабораторная работа 9. Объектно-реляционный подход к моделированию систем.

Лабораторная работа 10. Средства идентификации объектов в модели и средства моделирования

Лабораторная работа 11. Оптимизация запросов в базе данных

Тема 4.4. Промышленные системы управления базами данных. Обзор промышленных СУБД.

Классификация промышленных СУБД. Независимость, избыточность данных в СУБД. Физическая организация базы данных. Хешированные, индексированные файлы. Физические и логические базы данных. Логические структуры данных и структуры данных для приложений. Произвольная и последовательная выборки. Защита баз данных; целостность и сохранность баз данных. Перспективные направления и тенденции развития баз и банков данных.

Лабораторная работа 12. Хешированные, индексированные файлы

Лабораторная работа 13. Логические структуры данных и структуры данных для приложений.

5. ПЕРЕЧЕНЬ УЧЕБНО-МЕТОДИЧЕСКОГО ОБЕСПЕЧЕНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ «ТЕХНОЛОГИИ БАЗ ДАННЫХ»

- 1) Методическая система, используемая авторами данной рабочей программы, базируется на оптимальном сочетании активных форм и методов организации учебной деятельности студентов (лекция, беседа, анализ, синтез, мозговой штурм и т.п.), приемов групповой (выполнение и защита заданий по разработке алгоритмов и программ) и самостоятельной работы (разработка и защита программ).
- 2) Все студенты являются активными пользователями ресурса системы LMS MOODLE, поскольку в нем представлены конспекты всех лекций и методические разработки к проведению каждого практического и лабораторного занятия.
- 3) В течение всего периода обучения организуется регулярная проверка и учет выполнения домашних заданий, размещенных в LMS MOODLE.
- 4) Промежуточная аттестация принимается в форме экзамена по заранее определенному перечню вопросов с обязательным самостоятельным написанием кодов базовых алгоритмов и функций, разобранных и протестированных на занятиях любого вида в течение семестра.
- 5) Ресурс LMS MOODLE поддерживается в актуальном состоянии.
- 6) Одной из важнейших задач методического обеспечения учебной деятельности студента является формирование устойчивого навыка разработки алгоритмов и работы в среде программирования посредством использования ресурсов удаленного рабочего стола, расположенного на сервере кафедры, а также контроль умений студентов читать, анализировать и использовать в работе учебную и специальную литературу.
- 7) По дисциплине используется рейтинг.

6. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ

6.1. Перечень компетенций с указанием этапов их формирования в процессе освоения образовательной программы

Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения образовательной программы представлен в таблице пункта 1 рабочей программы.

Формирование компетенции «Способность использовать современные инструментальные и вычислительные средства» (ПК-3) осуществляется в три этапа. Первый этап формирования компетенции осуществляется в процессе освоения дисциплин «Операционные системы» и «Технологии

баз данных». Второй этап формирования компетенции осуществляется в процессе освоения дисциплин «Теория автоматов и формальных языков» и «Технологии баз данных». Третий этап формирования компетенции осуществляется в процессе прохождения производственной практики (научно-исследовательская работа).

6.2. Описание показателей, критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания

Дескриптор компетенций	Показатели оценивания	Критерии оценивания
Знания	основные понятия в области теории баз данных; основные технологии обработки баз данных; этапы жизненного цикла баз данных; виды и способы организации запросов к данным в реляционных моделях; способы программирования и администрирования баз данных.	Отметка «зачтено» выставляется, если студент в целом за семестр набрал от 41 до 100 баллов (с учетом баллов, набранных на промежуточной аттестации (зачете)). Отметка «не зачтено» выставляется, если студент в целом за семестр набрал менее 41 балла (с учетом баллов, набранных на промежуточной аттестации (зачете)).
Умения	выбирать технологию для проектирования реляционной модели данных; разрабатывать структуру модели данных; реализовывать этапы жизненного цикла баз данных; использовать технологии баз данных для обработки табличных данных; представлять отчеты по результатам обработки данных средствами СУБД.	Оценка «отлично» выставляется, если студент в целом за семестр набрал от 81 до 100 баллов (при условии, что на экзамене набрано не менее 20 баллов). Оценка «хорошо» выставляется, если студент в целом за семестр набрал от 61 до 80 баллов (при условии, что на экзамене набрано не менее 20 баллов). Оценка «удовлетворительно» выставляется, если студент в целом за семестр набрал от 41 до 60 баллов (при условии, что на экзамене набрано не менее 10 баллов).
Навыки и опыт деятельности	основными методами, способами и средствами получения, хранения, обработки информации в реляционных моделях данных; типовыми технологиями разработки баз данных	Оценка «неудовлетворительно» выставляется, если студент в целом за семестр набрал менее 41 балла (или на экзамене набрал менее 10 баллов).

Критерии оценивания компетенций формируются на основе балльно-рейтинговой системы с помощью всего комплекса методических материалов, определяющих процедуры оценивания знаний, умений, навыков и опыта деятельности, характеризующих данный этап формирования компетенций.

Баллы, набранные студентом в течение семестра	Баллы за промежуточную аттестацию (зачет)	Общая сумма баллов за модуль в семестр	Отметка
11 – 80	0 – 20	41-100	Зачтено
0 – 20	0 – 20	0 – 40	Не зачтено

Знания, умения, навыки и компетенции студентов по дисциплине оцениваются по двухбалльной шкале с отметками: «зачтено»; «не зачтено». Как правило, при двухбалльной системе преподавателями используются следующие показатели, при условии успешного прохождения текущего контроля успеваемости по дисциплине:

Оценка «зачтено» ставится, если студент освоил программный материал всех разделов, последователен в изложении программного материала, достаточно последовательно и логически стройно его излагает, умеет увязывать теорию с практикой, успешно прошел текущий контроль успеваемости по дисциплине, продемонстрировал индивидуальные знания, умениями и навыки практической работы.

Оценка «не зачтено» ставится, если студент не знает значительной части программного материала, допускает существенные ошибки, непоследователен в его изложении, не прошел текущий контроль успеваемости, не в полной мере владеет необходимыми знаниями, умениями и навыками при выполнении практических заданий, то есть студент не может продолжить обучение без дополнительной подготовки по соответствующей дисциплине.

Составляющие итоговой оценки за дисциплину:

1) Текущий контроль (общий вес 60 баллов):

до 21 балла – посещение занятий;

до 26 баллов – выполнение заданий в ходе выполнения лабораторных работ в LMS Moodle и заданий для самостоятельной работы

до 10 баллов – выполнение отдельно выделенных в методических указаниях к выполнению проектного задания

до 3 баллов – выполнение контрольной работы в аудитории

2) Итоговый контроль заключается в проведении экзамена (общий вес - 40 баллов). Экзамен проводится по вопросам с обязательным решением задач. Как правило, студент получает два вопроса из приведенного выше списка и одну задачу на составление запроса, готовится в присутствии преподавателя и дает подробные комментарии. Студент, пропускавший занятия в ходе семестра, получает дополнительные вопросы и задачи по каждой пропущенной им теме (на усмотрение преподавателя).

Для получения положительной итоговой оценки на экзамене необходимо получить не менее 50% по каждой составляющей и выполнить все лабораторные работы. Шкала перевода баллов в оценку приведена в таблице.

Оценка	«отлично»	«хорошо»	«удовлетворительно»	«не удовлетворительно»
Интервал количества баллов	81..100	61..80	41..60	0..40

6.3. Типовые контрольные задания и иные материалы, характеризующие этапы формирования компетенций в процессе освоения образовательной программы

Вопросы к зачету

1. Основные компоненты СУБД.
2. Преимущества и недостатки СУБД.
3. Жизненный цикл СУБД.
4. Проектирование баз данных.
5. Моделирование данных. Определение связей между объектами.
6. Реляционная модель данных. Логическое проектирование БД.
7. Операции над реляционными отношениями.
8. Нормализация отношений.
9. Обеспечение целостности данных.

10. Функции СУБД.
11. Этапы развития БД.
12. Обзор наиболее популярных локальных (настольных) СУБД.
13. Серверные СУБД. Основные черты современных серверных СУБД.
14. Понятие СУБД, иерархические, сетевые, реляционные и объектно-ориентированные базы данных. Основные компоненты СУБД. Преимущества и недостатки СУБД.
15. Этапы развития технологий представления БД и СУБД.
16. Необходимость СУБД. Перспективы развития СУБД.
17. Жизненный цикл базы данных. Выбор системы управления базами данных. Функции СУБД.
18. СУБД с централизованной архитектурой, архитектурой файл-сервер, клиент-сервер и трехуровневой архитектурой Тонкий клиент - Сервер приложений - Сервер базы данных.
19. Физические модели организации баз данных.
20. Сетевые и иерархические СУБД.
21. Реляционная алгебра.
22. Реляционная модель данных: таблицы, первичные и внешние ключи, реализация отношений "предок-потомок".
23. Нормализация отношений. Обеспечение целостности данных.
24. Проектирование баз данных. Моделирование данных. Определение связей между объектами.
25. Модель данных "сущность"- "связь". CASE-средства проектирования БД.
26. ER-диаграммы. Виды связей и отношений.
27. Переход от ER – модели к реляционной.
28. Реляционная модель данных. Логическое проектирование БД. Реляционная алгебра.
29. OLTP и OLAP системы. Назначение и основные характеристики.
30. Распределенные БД.
31. Организация процессов обработки данных в БД. Ограничения целостности.

Вопросы к экзамену

1. Язык SQL. Основные понятия. Механизм обработки запросов.
2. Язык SQL. Предложение SELECT.
3. Язык SQL. Предложение INSERT.
4. Язык SQL. Предложение UPDATE.
5. Язык SQL. Предложение DELETE.
6. Язык SQL. Выборка данных по логическому условию.
7. Язык SQL. Агрегатные функции.
8. Язык XML. Основы технологии. Назначение и области применения. XML-серверы.
9. Языковые средства современных СУБД.
10. СУБД MySQL: обзор возможностей, структура БД MySQL, язык SQL.
11. Основные компоненты MySQL.
12. Репликация, архивация и резервное копирование данных средствами MySQL.
13. Информационные и служебные функции для работы с базой данных и таблицами данных в MySQL.
14. Обзор коммерческих СУБД: Oracle, MS SQL Server, Informix Universal Server, PostgreSQL, DB2, Visual FoxPro, MS Access – различия в архитектуре, возможностях и области применения.
15. Язык SQL – место SQL в схеме взаимодействия с базой данных, возможности и преимущества SQL.
16. Запросы, невыразимые средствами реляционной алгебры. Ограничения SQL.
17. Механизм обработки SQL-запросов. Оптимизация выполнения SQL-запросов.
18. Типы данных в SQL. Преобразование данных. Оператор CAST.
19. Статические и динамические SQL запросы.
20. Простые запросы в SQL: основные конструкции, вычисляемые столбцы, условия отбора, сортировка результатов, объединение результатов нескольких запросов.

21. Многотабличные запросы в SQL: запросы на объединение двух и более таблиц, внутренние и внешние объединения.
22. Итоговые запросы в SQL: статистические функции, запросы с группировкой, условия отбора групп, подчиненные запросы.
23. Комбинированные запросы. Использование оператора UNION.
24. Использование подзапросов. Использование операторов EXISTS, ANY, ALL и SOME.
25. Создание представлений. Использование оператора CREATE VIEW.
26. Изменение данных при помощи SQL: добавление, удаление, обновление данных. Псевдонимы и индексы таблиц.
27. Создание и удаление таблиц в SQL, целостность данных: правила удаления и обновления данных, ограничения на значение, домены, утверждения.
28. Обработка транзакций: понятие транзакции, журнал транзакций, транзакции и работа в многопользовательском режиме, блокировка.
29. Система безопасности данных в SQL: идентификаторы пользователей, защищаемые объекты, привилегии, предоставление и отмена привилегий.
30. Хранимые процедуры и функции – понятие, синтаксис в SQL. Реализация в MySQL.
31. Триггеры – понятие, синтаксис в SQL. Реализация в MySQL.
32. Экспорт и импорт данных в MySQL.
33. Web-ориентированные базы данных. Реализация интерфейса с БД MySQL с помощью скриптовых языков.
34. Взаимодействие PHP и MySQL. Установка и завершение соединения. Обработка запросов.
35. Основные функции PHP для работы с MySQL.
36. Аутентификация пользователей средствами PHP и MySQL.
37. DataMining. Основные задачи и процессы DataMining.
38. Задачи классификации и кластеризации в DataMining.
39. Обзор программных продуктов в сфере DataMining.

Контроль промежуточных результатов обучения проводится в процессе выполнения лабораторных работ и проектирования баз данных по индивидуальным проектам.

Результаты обучения	Основные показатели результатов подготовки	Формы и методы контроля
Понимать сущность и социальную значимость своей будущей профессии, проявлять к ней устойчивый интерес	- демонстрация интереса к будущей профессии	Экспертное наблюдение и оценка на практических занятиях, выполнения работ по производственной практике
Организовывать собственную деятельность, выбирать типовые методы и способы выполнения профессиональных задач, оценивать их эффективность и качество	- обоснование выбора и применения методов и способов решения профессиональных задач в области разработки технологических процессов; - демонстрация эффективности и качества выполнения профессиональных задач	Устный опрос Экспертное наблюдение и оценка на практических занятиях, выполнения работ по производственной практике
Принимать решения в стандартных и нестандартных ситуациях и нести за них ответственность	- демонстрация способности принимать решения в стандартных и нестандартных ситуациях и нести за них ответственность	Экспертное наблюдение и оценка на практических занятиях, выполнения самостоятельных работ и заданий по производственной практике

Осуществлять поиск и использование информации, необходимой для эффективного выполнения профессиональных задач, профессионального и личностного развития	- нахождение и использование информации для эффективного выполнения профессиональных задач, профессионального и личностного развития	Экспертное наблюдение и оценка на практических занятиях, выполнения самостоятельных работ и заданий по производственной практике
Использовать информационно-коммуникационные технологии в профессиональной деятельности	- демонстрация навыков использования информационно-коммуникационные технологии в профессиональной деятельности	Экспертное наблюдение и оценка на практических занятиях, выполнения работ по производственной практике
Работать в коллективе и в команде, эффективно общаться с коллегами, руководством, потребителями	- взаимодействие с обучающимися, преподавателями и мастерами в ходе обучения	Экспертное наблюдение и оценка на практических занятиях, выполнения работ по производственной практике
Брать на себя ответственность за работу членов команды (подчиненных), результат выполнения заданий	- проявление ответственности за работу подчиненных, результат выполнения заданий	Экспертное наблюдение и оценка на практических занятиях, выполнения работ по производственной практике
Самостоятельно определять задачи профессионального и личностного развития, заниматься самообразованием, осознанно планировать повышение квалификации	- планирование обучающимся повышения личностного и квалификационного уровня	Экспертное наблюдение и оценка на практических занятиях, выполнения самостоятельных работ и заданий по производственной практике
Ориентироваться в условиях частой смены технологий в профессиональной деятельности	- проявление интереса к инновациям в области профессиональной деятельности	Экспертное наблюдение и оценка на практических занятиях, выполнения самостоятельных работ и заданий по производственной практике
Использовать информационно-коммуникационные технологии в профессиональной деятельности	- демонстрация навыков использования информационно-коммуникационные технологии в профессиональной деятельности	Экспертное наблюдение и оценка на практических занятиях, выполнения работ по производственной практике

Примерный список заданий на проектирование модели данных в соответствии с особенностями предметной области

1. Решение заданий, использующих объекты базы данных, предполагает выполнение следующих этапов:
2. 1. Проектирование инфологической модели задачи. Определение сущностей, атрибутов сущностей, идентифицирующих атрибутов, связей между сущностями. При проектировании должны учитываться требования гибкости структур для выполнения перечисленных функций и не избыточного хранения данных.
3. 2. Проектирование схемы базы данных: описание схем таблиц, типов (доменов) атрибутов, определение ограничений целостности.
4. 3. Реализация запросов, указанных в задаче.

В заданиях дана некоторая спецификация решаемой задачи. Спецификация не предполагает оптимального определения структур данных, но задает полный перечень необходимой и хранимой в базе данных информации и выполняемых программой функций.

1. База данных хроники восхождений в альпинистском клубе. 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. Дайте выразительные имена таблицам и полям, в которые могла бы заноситься указанная информация.

Написать пакет, состоящий из процедур и функций, которые позволили бы выполнить следующие действия с базой данных: 1) для каждой горы показать список групп, осуществлявших восхождение, в хронологическом порядке; 2) предоставить возможность добавления новой вершины, с указанием названия вершины, высоты и страны местоположения; 3) предоставить возможность изменения данных о вершине, если на нее не было восхождения; 4) показать список альпинистов, осуществлявших восхождение в указанный интервал дат; 5) предоставить возможность добавления нового альпиниста в состав указанной группы; 6) показать информацию о количестве восхождений каждого альпиниста на каждую гору; 7) показать список восхождений (групп), которые осуществлялись в указанный пользователем период времени; 8) предоставить возможность добавления новой группы, указав ее название, вершину, время начала восхождения; 9) предоставить информацию о том, сколько альпинистов побывали на каждой горе. Предусмотреть разработку триггеров, обеспечивающих каскадные изменения в связанных таблицах.

2. База данных медицинского кооператива. Базу данных использует для работы коллектив врачей. В таблицы должны быть занесены 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, в таблицу заносится название лекарства, способ его приема, словесное описание предполагаемого действия и возможных побочных эффектов.

Создать пакет, состоящий из функций и процедур, позволяющих: 1) по заданной дате определить количество вызовов в этот день; 2) позволяют определить количество больных, заболевших данной болезнью; 3) по заданному лекарству определить его побочный эффект порядке; 4) предоставить возможность добавления нового лекарства с описанием его свойств в БД. Предусмотреть разработку триггеров, обеспечивающих каскадные изменения в связанных таблицах.

3. База данных Городской Думы. В базе хранятся имена, адреса, домашние и служебные телефоны всех членов Думы. В Думе работает порядка сорока комиссий, все участники которых являются членами Думы. Каждая комиссия имеет свой профиль, например, вопросы образования, проблемы, связанные с жильем и так далее. Данные по каждой из комиссий включают: ее нынешний состав и председатель, прежние председатели и члены этой комиссии, участвовавшие в ее работе за прошедшие 10 лет, даты включения и выхода из состава комиссии, избрания ее председателей. Члены Думы могут заседать в нескольких комиссиях. В базу заносятся время и место проведения каждого заседания комиссии с указанием депутатов и служащих Думы, которые участвуют в его организации.

Создать пакет с процедурами и функциями, которые позволяют выполнять следующие действия: 1) показать список комиссий, для каждой ее состав и председателя; 2) предоставить возможность добавления нового члена комиссии; 3) показать список членов муниципалитета, для каждого из них список комиссий, в которых он участвовал и/или был председателем; 4) предоставить возможность добавления новой комиссии, с указанием председателя; 5) для указанного интервала дат и комиссии выдать список ее членов с указанием количества пропущенных заседаний; 6) предоставить возможность добавления нового заседания, с указанием присутствующих; 7) по каждой комиссии показать количество проведенных заседаний в указанный период времени. Предусмотреть разработку триггеров, обеспечивающих каскадные изменения в связанных таблицах.

4. База данных рыболовной фирмы. Фирме принадлежит небольшая флотилия рыболовных катеров. Каждый катер имеет паспорт, куда занесены его название, тип, водоизмещение и дата по-

стройки. 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 не взвешивается.

Написать запросы, осуществляющие следующие операции: 1) для каждого катера вывести даты выхода в море с указанием улова; 2) предоставить возможность добавления выхода катера в море с указанием команды; 3) для указанного интервала дат вывести для каждого сорта рыбы список катеров с наибольшим уловом; 4) для указанного интервала дат вывести список банок, с указанием среднего улова за этот период; 5) предоставить возможность добавления новой банки с указанием данных о ней; 6) для заданной банки вывести список катеров, которые получили улов выше среднего; 7) вывести список сортов рыбы и для каждого сорта список рейсов с указанием даты выхода и возвращения, количества улова; 8) для выбранного пользователем рейса и банки добавить данные о сорте и количестве пойманной рыбы; 9) предоставить возможность пользователю изменять характеристики выбранного катера; 10) предоставить возможность добавления нового катера; 11) для указанного сорта рыбы и банки вывести список рейсов с указанием количества пойманной рыбы. Предусмотреть разработку триггеров, обеспечивающих каскадные изменения в связанных таблицах.

5. База данных фирмы, проводящей аукционы. Фирма занимается продажей с аукциона антикварных изделий и произведений искусства. Владельцы вещей, выставяемых на проводимых фирмой аукционах, юридически являются продавцами. Лица, приобретающие эти вещи, именуются покупателями. Получив от продавцов партию предметов, фирма решает, на котором из аукционов выгоднее представить конкретный предмет. Перед проведением очередного аукциона каждой из выставяемых 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, выставяются картины, написанные маслом и не ранее 1900 г.). Заносятся также сведения о каждом продаваемом предмете: аукцион, на который он заявлен, номер лота, продавец, отправная цена и краткое словесное описание. Продавцу разрешается выставять 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.

Создать пакет, состоящий из процедур и функций, позволяющий осуществить следующие операции: 1) для указанного интервала дат вывести список аукционов с указанием наименования, даты и места проведения; 2) добавить на указанный пользователем аукцион на продажу предмет искусства с указанием начальной цены; 3) вывести список аукционов, с указанием суммарного дохода от продажи, отсортированных по доходу; 4) для указанного интервала дат, вывести список предметов, которые были проданы на аукционах в этот период времени; 5) предоставить возможность добавления факта продажи на указанном аукционе заданного предмета; 6) для указанного интервала дат вывести список продавцов с указанием общей суммы, полученной от продажи предметов в этот промежуток времени; 7) вывести список покупателей, которые сделали приобретения в указанный интервал дат; 8) предоставить возможность добавления записи о проводимом аукционе (место, время); 9) для указанного места, вывести список аукционов; 10) для указанного интервала дат вывести список продавцов, которые принимали участие в аукционах, проводимых в этот период времени; 11) предоставить возможность добавления и изменения информации о продавцах и покупателях; 12) вывести список покупателей с указанием количества приобретенных предметов в указанный период времени. Предусмотреть разработку триггеров, обеспечивающих каскадные изменения в связанных таблицах.

6. База данных библиотеки. Разработать информационную систему обслуживания библиотеки, которая содержит следующую информацию: названия книг, ФИО авторов, наименования издательств, год издания, количество страниц, количество иллюстраций, стоимость, название филиала библиотеки или книгохранилища, 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.

Необходимо составить пакет из процедур и функций, который позволяет: 1) для указанного филиала посчитать количество экземпляров указанной книги, находящихся в нем; 2) для указанной книги посчитать количество факультетов, на которых она используется в данном филиале, и вывести названия этих факультетов; 3) предоставить возможность добавления и изменения информации о книгах в библиотеке; 4) предоставить возможность добавления и изменения информации о филиалах; 5) предусмотреть разработку триггеров, срабатывающих на пользовательские исключительные ситуации; Предусмотреть разработку триггеров, обеспечивающих каскадные изменения в связанных таблицах. 7. 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 или название группы); об учебном плане (название специальности, дисциплина, семестр, количество отводимых на дисциплину часов, форма отчетности (экзамен/зачет)); о журнале успеваемости студентов (год/семестр, студент, дисциплина, оценка). Разработать пакет, состоящий из процедур и функций, позволяющий: 1) для указанной формы обучения посчитать количество студентов этой формы обучения; 2) для указанной дисциплины получить количество часов и формы отчетности по этой дисциплине; предоставить возможность добавления и изменения информации о студентах, об учебных планах, о журнале успеваемости при этом предусмотреть курсоры, срабатывающие на некоторые пользовательские исключительные ситуации; 3) предоставить возможность добавления и изменения информации о журнале успеваемости. Предусмотреть разработку триггеров, обеспечивающих каскадные изменения в связанных таблицах.

8. База данных для учета аудиторного фонда университета. База данных должна содержать следующую информацию об аудиторном фонде университета. Наименование корпуса, в котором расположено помещение, номер комнаты, расположение комнаты в корпусе, ширина и длина комнаты в метрах, назначение и вид помещения, подразделение университета, за которым закреплено помещение. В базе данных также должна быть информация о высоте потолков в помещениях в зависимости от места расположения помещений в корпусе. Следует также учитывать, что структура подразделений университета имеет иерархический вид, когда одни подразделения входят в состав других (факультет, кафедра, лаборатория).

Помимо SQL запросов для создания таблиц базы данных, разработать пакет, состоящий из процедур и функций, позволяющий: 1) рассчитать данные о площадях и объемах каждого помещения; 2) для указанного корпуса получить количество факультетов, их названия и структуру, находящиеся в этом корпусе; 3) предоставить возможность добавления и изменения информации о корпусах в университете, при этом предусмотреть курсоры, срабатывающие на некоторые пользовательские исключительные ситуации; 4) предоставить возможность добавления и изменения информации о комнатах в корпусах университета, при этом предусмотреть курсоры, срабатывающие на некоторые пользовательские исключительные ситуации. Предусмотреть разработку триггеров, обеспечивающих каскадные изменения в связанных таблицах.

9. База данных для регистрации происшествий. Необходимо создать Базу данных для регистрации происшествий. База данных должна содержать данные для регистрации сообщений о происшествиях (регистрационный номер сообщения, дата регистрации, краткая фабула (тип происшествия)); информацию о принятом по происшествию решении (отказано в возбуждении дел, удовлетворено ходатайство о возбуждении уголовного дела с указанием регистрационный номера заведенного дела, отправлено по территориальному признаку); информацию о лицах, виновных или

подозреваемых в совершении происшествия (регистрационный номер лица, фамилия, имя, отчество, адрес, количество судимостей), отношение конкретных лиц к конкретным происшествиям (виновник, потерпевший, подозреваемый, свидетель).

Помимо SQL запросов для создания таблиц базы данных, разработать пакет, состоящий из процедур и функций, позволяющий: 2) рассчитать данные о количестве происшествий в указанный промежуток времени; 3) для указанного лица получить количество происшествий, в которых он зарегистрирован; 4) предоставить возможность добавления и изменения информации о происшествиях, при этом предусмотреть курсоры, срабатывающие на некоторые пользовательские исключительные ситуации; 5) предоставить возможность добавления и изменения информации о лицах, участвующих в происшествиях, при этом предусмотреть курсоры, срабатывающие на некоторые пользовательские исключительные ситуации. Предусмотреть разработку триггеров, обеспечивающих каскадные изменения в связанных таблицах.

10. База данных для обслуживания работы конференции. База данных должна содержать справочник персоналий участников конференции (фамилия, имя, отчество, ученая степень, ученое звание, научное направление, место работы, кафедра (отдел), должность, страна, город, почтовый индекс, адрес, рабочий телефон, домашний телефон, e-mail), и информацию, связанную с участием в конференции (докладчик или участник, дата рассылки 1-го приглашения, дата поступления заявки, тема доклада, от-метка о поступлении тезисов, дата рассылки 2-го приглашения, дата поступления оргвзноса, размер поступившего оргвзноса, дата приезда, дата отъезда, потребность в гостинице).

Помимо SQL запросов для создания таблиц базы данных, разработать пакет, состоящий из процедур и функций, позволяющий: 1) для указанной даты 1-ой рассылки вывести список приглашенных и посчитать их количество; 2) предоставить возможность добавления приглашенных на конференцию с указанием оргвзноса и даты его уплаты; 3) вывести список приглашенных, с указанием даты об уплате оргвзноса; 4) для указанной интервала дат, вывести список участников, уплативших оргвзнос в этом диапазоне; 5) для указанного города вывести название тезисов докладов, поступивших из этого города; 6) для указанного города, вывести список нуждающихся в гостинице. Предусмотреть разработку триггеров, обеспечивающих каскадные изменения в связанных таблицах.

11. База данных для обслуживания склада. База данных должна обеспечить автоматизацию складского учета. В ней должны содержаться следующие данные: информация о единицах хранения, номер ордера, дата, код поставщика, балансный счет, код сопроводительного документа по справочнику документов, номер сопроводительного документа, код материала по справочнику материалов, счет материала, код единицы измерения, количество пришедшего материала, цена единицы измерения; информация о хранящихся на складе материалах (справочник материалов: код класса материала, код группы материала, наименование материала); информация о единицах измерения конкретных видов материалов, код материала, единица измерения (метры, килограммы, литры и т.д.); информация о поставщиках материалов, код поставщика, его наименование, ИНН, юридический адрес (индекс, город, улица, дом), адрес банка (индекс, город, улица, дом), номер банковского счета.

Помимо SQL запросов для создания таблиц базы данных, разработать пакет, состоящий из процедур и функций, позволяющий: 1) посчитать количество поставщиков данного материала; 2) предоставить возможность добавления единицы хранения с указанием всех реквизитов; 3) вывести список поставщиков с указанием всех реквизитов данного материала на склад; 4) для указанного адреса банка посчитать количество поставщиков склада, пользующихся услугами этого банка. Предусмотреть разработку триггеров, обеспечивающих каскадные изменения в связанных таблицах.

12. База данных фирмы. Фирма отказалась от приобретения некоторых товаров у своих поставщиков, решив самостоятельно наладить их производство. С этой целью она организовала сеть специализированных цехов, каждый из которых принимает определенное участие в технологическом процессе. Каждому виду выпускаемой продукции присваивается, как обычно, свой шифр товара, под которым он значится в файле товарных запасов. Этот же номер служит и шифром продукта. В записи с этим шифром указывается, когда была изготовлена последняя партия этого продукта, какова ее стоимость, сколько операций потребовалось. Операцией считается законченная часть процесса производства, которая целиком выполняется силами одного цеха в соответствии с техническими требованиями, перечисленными на отдельном чертеже. Для каждого продукта и для каждой операции в базе данных фирмы заведена запись, содержащая описание операции, ее среднюю продолжительность и номер чертежа, по которому можно отыскать требуемый чертеж. Кроме того, указывается номер цеха, обычно производящего данную операцию. В запись, связанную с конкретной операцией, заносятся потребные количества расходуемых материалов, а также присвоенные им шифры товара. Расходуемыми называют такие материалы, как, например, электрический кабель, который нельзя использовать повторно. Когда, готовясь к выполнению операции, расходуемый материал забирают со склада, регистрируется фактически выданное количество, соответствующий шифр товара, номер служащего, ответственного за выдачу, дата и время выдачи, номер операции и номер наряда на проведение работ, который будет обсуждаться ниже. Реально затраченное количество материала может не совпадать с расчетным, из-за того, например, что часть изготовленной продукции бракуется. Каждый из цехов располагает многочисленными инструментами и приспособлениями. При выполнении некоторых операций их все же не хватает, и цех вынужден обращаться в центральную инструментальную за недостающими. Каждый тип инструмента снабжен отдельным номером и на него заведена запись со словесным описанием. Кроме того, там отмечено, какое количество инструментов этого типа выделено цехам и какое осталось в инструментальной. Экземпляры инструмента конкретного типа, например гаечные ключи одного размера, различаются по своим индивидуальным номерам. На фирме для каждого типа инструмента имеется запись, содержащая перечень всех индивидуальных номеров. Кроме того, указаны даты их поступления на склад. По каждой операции в фирме отмечают типы и количества инструментов этих типов, которые должны использоваться при ее выполнении. Когда инструменты действительно берутся со склада, фиксируется индивидуальный номер каждого экземпляра, указываются номер заказавшего их цеха и номер наряда на проведение работ. И в этом случае затребованное количество не всегда совпадает с заказанным. Наряд на проведение работ по форме напоминает заказ на приобретение товаров, но, в отличие от последнего, он направляется не поставщику, а в один из цехов. Оформляется этот наряд после того, как руководство фирмы сочтет необходимым выпустить партию некоторого продукта. В наряд заносятся шифр продукта, дата оформления наряда, срок, к которому должен быть выполнен заказ, а также требуемое количество продукта. Разработайте структуру таблиц базы данных, подберите имена таблиц и полей, в которых могла бы разместиться вся эта информация.

Помимо SQL запросов для создания таблиц базы данных, разработать пакет, состоящий из процедур и функций, позволяющий: 1) для выбранного цеха, выдать список операций, выполняемых им. Для каждой операции список расходуемых материалов, с указанием количества; 2) показать список инструментов и предоставить возможность добавления нового; 3) выдать список используемых инструментов; 4) для указанного интервала дат, вывести список нарядов; 5) показать список операций и предоставить возможность добавления новой операции; 6) выдать список расходуемых материалов, используемых в различных нарядах; 7) выдать список товаров, с указанием используемых инструментов; 8) показать список нарядов и предоставить возможность добавления нового; 9) выдать отчет о производстве товаров различными цехами, указав наименование цеха, название товара и его количество. Предусмотреть разработку триггеров, обеспечивающих каскадные изменения в связанных таблицах.

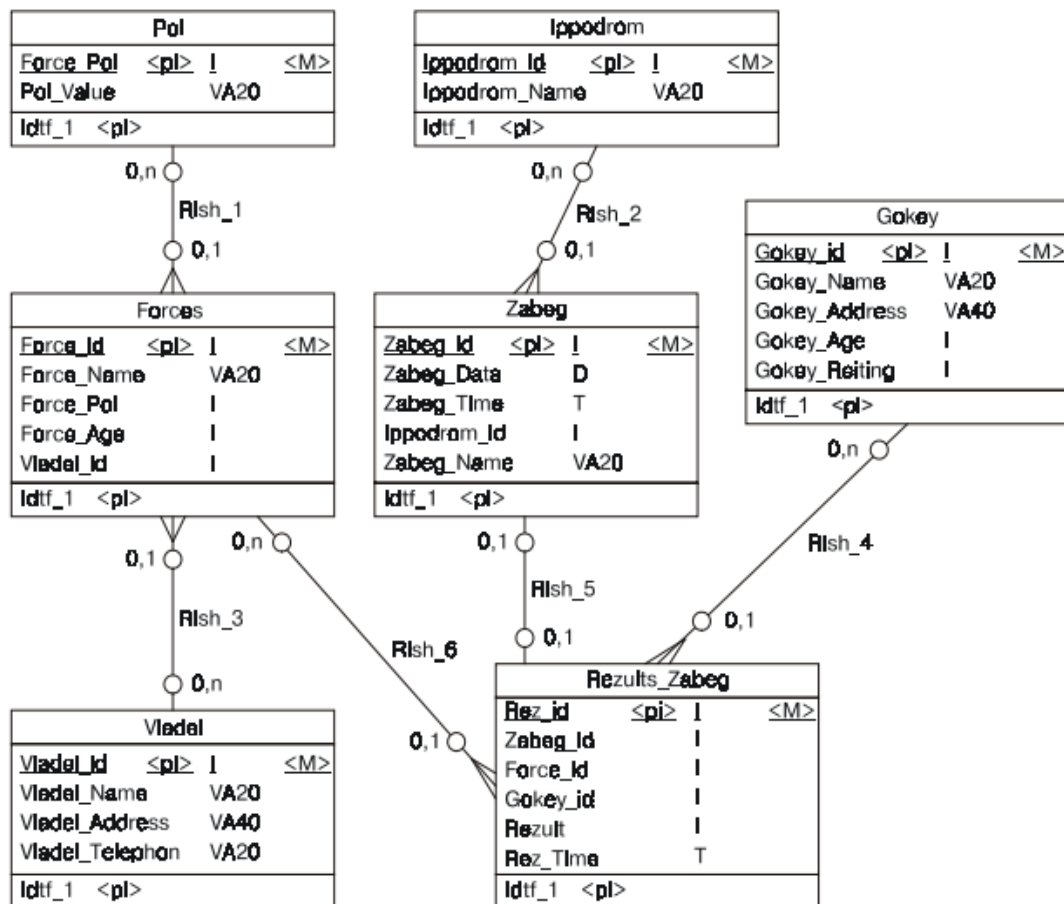
13. База данных музыкального магазина. Таблицы базы данных содержат информацию о музыкантах, музыкальных произведениях и обстоятельствах их исполнения. Несколько музыкантов, образующих единый коллектив, называются ансамблем. Это может быть классический оркестр, джазовая группа, квартет, квинтет и т.д. 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. Каждая пластинка, а точнее, ее на-клейка, идентифицируется отдельным номером, так что всем копиям, отпечатанным с матрицы в разное время, присвоены одинаковые номера. На пластинке может быть записано несколько исполнений одного и того же произведения ó для каждого из них в базе заведена отдельная запись. Когда выходит новая пластинка, регистрируется название выпускавшей ее компании (например, EMI), 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.

Помимо SQL запросов для создания таблиц базы данных, раз-работать пакет, состоящий из процедур и функций, позволяющий: 1) количество музыкальных произведений заданного ансамбля; 2) выводит название всех компакт-дисков заданного ансамбля; 3) показать лидеров продаж текущего гола, то есть названия компакт-дисков, которые чаще всего покупали в текущем году; 4) предусмотреть изменения данных о компакт-дисках и ввод новых данных; 5) предусмотреть ввод новых данных об ансамблях. Предусмотреть разработку триггеров, обеспечивающих каскадные изменения в связанных таблицах.

Пример выполнения лабораторной работы

1. **Постановка задачи.** База данных Скачки. В информационной системе клуба любителей скачек должна быть представлена информация об участвующих в скачках лошадях (кличка, пол, возраст), их владельцах (имя, адрес, телефон) и жокеях (имя, адрес, возраст, рейтинг). Необходимо сформировать таблицы для хранения информации по каждому состязанию: дата, время и место проведения скачек (ипподром), название состязаний (если таковое имеется), клички участвующих в заездах лошадей и имена жокеев, занятые ими места и показанное в заезде время.
2. Требуется: • сформировать структуру таблиц базы данных; • подобрать подходящие имена таблицам и их полям; • обеспечить требования нормализации таблиц базы данных; • сформировать SQL-запросы для создания таблиц базы данных с указанием первичных и внешних ключей и требуемых ограничений; • создать пакет с курсором и триггеры.
3. **Описание структуры таблиц базы данных.** Опишем структуру таблиц, связи между ними и произведем их нормализацию. Создадим две вспомогательные таблицы Pol и Ippodrom.
4. Таблица Pol содержит возможные значения пола лошади (поле Pol_Value) и их порядковые номера (поле Force_Pol). Поле Force_Pol является первичным ключом таблицы. Force_Pol Pol_Value 1 мужской 2 женский
5. Таблица Ippodrom содержит возможные названия ипподромов (поле Ippodrom_Name) и их порядковые номера (поле Ippodrom_id). Поле Ippodrom_id является первичным ключом таблицы. Ippodrom_id Ippodrom_Name
6. Таблица Forces содержит информацию о лошадях. Эта таблица содержит следующие поля. Force_id – содержит идентификатор лошади и это поле объявлено как первичный ключ. Force_Name – содержит имя лошади и является уникальным. Force_Pol – содержит информацию о поле лошади и является внешним ключом (оно ссылается на поле Force_Pol таблицы Pol).
7. Таблица Gokey Force_Age – содержит информацию о возрасте лошади. Vldel_id – содержит информацию о владельце лошади и является внешним ключом (ссылается на поле Vldel_id таблицы Vldel).

8. Таблица *Gokey* содержит информацию о жокеях. Эта таблица содержит следующие поля. *Gokey_id* – содержит идентификатор жокея и это поле объявлено как первичный ключ. *Gokey_Name* – содержит имя жокея. *Gokey_Address* – содержит адрес жокея. *Gokey_Age* – содержит информацию о возрасте жокея и имеет ограничение ($Gokey_Age \geq 18$). *Gokey_Reiting* – содержит информацию о рейтинге жокея и имеет ограничение ($Gokey_Reiting \geq 0$).
9. Таблица *Vladel* содержит информацию о владельцах лошадей. Эта таблица содержит следующие поля. *Vladel_id* – содержит идентификатор владельца и это поле объявлено как первичный ключ. *Vladel_Name* – содержит имя владельца. *Vladel_Address* – содержит адрес владельца. *Vladel_Telephon* – содержит телефон владельца.
10. Таблица *Zabeg* содержит информацию о забегах. Эта таблица содержит следующие поля. *Zabeg_id* – содержит идентификатор забега и это поле объявлено как первичный ключ. *Zabeg_Data* – содержит дату забега. *Zabeg_Time* – содержит время забега. *Ippodrom_id* – содержит идентификатор ипподрома и является внешним ключом (ссылается на поле *Ippodrom_id* таблицы *Ippodrom*). *Zabeg_Name* – содержит информацию о названии забега.
11. Таблица *Rezults_Zabeg* содержит результаты забегов. Эта таблица содержит следующие поля. *Rez_id* – содержит идентификатор результата забега и это поле объявлено как первичный ключ. *Zabeg_id* – содержит идентификатор забега и является внешним ключом (ссылается на поле *Zabeg_id* таблицы *Zabeg*). *Force_id* – содержит идентификатор лошади и является внешним ключом (ссылается на поле *Force_id* таблицы *Forces*). *Gokey_id* – содержит идентификатор жокея и является внешним ключом (ссылается на поле *Gokey_id* таблицы *Gokey*). *Rezult* – содержит информацию о занятом месте. *Rez_Time* содержит информацию о показанном в заезде времени.
12. **Концептуальная модель** базы данных приведена на Рисунке 1:



13.

14. Рисунок 1 Концептуальная модель базы данных

15. **Создание и заполнение таблиц.** Во всех таблицах изменения родительского ключа разрешаются, но при этом осуществляется коррекция всех значений внешних ключей, ссылающихся на модифицируемое значение родительского ключа. Таким образом обеспечивается ссылочная целостность базы данных. База данных находится в нормализованном состоянии.

16. **Создание и заполнение таблиц.**

17. **Таблица Пол:**

```
18. create table Pol (force_pol integer primary key, pol_value
    varchar(10) not null);
```

19.

20. **Заполнение таблицы:**

```
21. insert into Pol values (1, 'man'); insert into Pol values
    (2, 'woman'); Таблица ипподромы: create table ippodrom (ippodrom_id
    integer primary key, ippodrom_name varchar(20) unique not null);
    Заполнение таблицы: insert into ippodrom values (1, 'Chesmenka'); in-
    sert into ippodrom values (2, 'Sportsmen'); insert into ippodrom val-
    ues (3, 'Professional');
```

22. **Таблица Лошади:**

```
23. create table forces (force_id integer primary key, force_name
    varchar(20) unique not null, force_pol integer, force_age integer
    not null, vladel_id integer, FOREIGN KEY (force_pol) REFERENCES pol,
    FOREIGN KEY (vladel_id) REFERENCES vladel);
```

24. **Заполнение таблицы:**

```
25. insert into forces values (1, 'Veterok', 1, 3, 1); insert into forces
    values (2, 'Strela', 2, 2, 4); insert into forces values
    (3, 'Nochka', 2, 1, 3); insert into forces values (4, 'Pobeditel', 1, 2, 2);
    insert into forces values (5, 'Mig', 1, 1, 1); insert into forces values
    (6, 'Vasilinka', 2, 3, 2);
```

26. **Таблица Жокеи:**

```
27. create table gokey (gokey_id integer primary key, gokey_name
    varchar(20) not null, gokey_address varchar(40) not null, gokey_age
    integer check (gokey_age >= 18), gokey_reiting integer check
    (gokey_reiting >= 0));
```

28. **Заполнение таблицы:**

```
29. insert into gokey values (1, 'Ivlev', 'Moskva', 21, 314); insert into
    gokey values (2, 'Semenov', 'Moskva', 35, 245); insert into gokey values
    (3, 'Nikitin', 'Voronegh', 25, 450); insert into gokey values
    (4, 'Shevelev', 'Moskva', 28, 190); insert into gokey values
    (5, 'Popov', 'Krasnodar', 33, 254); insert into gokey values
    (6, 'Krasnov', 'SPB', 23, 120);
```

30. **Таблица Владельцы:**

```
31. create table vladel (vladel_id integer primary key, vladel_name
    varchar(20) not null, vladel_address varchar(40) not null,
    vladel_telephon varchar(20) not null);
```

32. **Заполнение таблицы:**

```
33. insert into vladel values (1, 'Petrov', 'Moskva', 1652763); insert
    into vladel values (2, 'Sidorov', 'SPB', 6525622); insert into vladel
    values (3, 'Semenov', 'Minsk', 7657652); insert into vladel values
    (4, 'Krasnova', 'Samara', 8766632);
```

34. Таблица Забеги:

```
35. create table zabeg (zabeg_id integer primary key, zabeg_data date
not null, zabeg_time varchar(20) not null, ippodrom_id integer,
zabeg_name varchar(20), FOREIGN KEY (ippodrom_id) REFERENCES
ippodrom);
```

36. Заполнение таблицы:

```
37. insert into zabeg values (1,'1-APR-02','18:30',1,'Aprelski'); in-
sert into zabeg values (2,'3-MAY-02','12:30',3,'Mayski'); insert in-
to zabeg values (3,'1-SEP-02','14:00',2,'Septem'); insert into zabeg
values (4,'1-DEC-02','18:30',1,'NewYear'); insert into zabeg values
(5,'1-FEB-03','18:30',2,'February'); insert into zabeg values (6,'15-
APR-03','11:00',3,'Aprelski03'); insert into zabeg values (7,'1-OCT-
03','18:30',1,'October03');
```

38. Таблица Результаты забегов:

```
39. create table results_zabeg (rez_id integer primary key, zabeg_id
integer, force_id integer, gokey_id integer, result integer not
null, rez_time real not null, FOREIGN KEY(force_id) REFERENCES forc-
es, FOREIGN KEY(gokey_id) REFERENCES gokey, FOREIGN KEY(zabeg_id)
REFERENCES zabeg);
```

40. Заполнение таблицы:

```
41. insert into results_zabeg values (1,1,1,2,1,12.23); insert into
results_zabeg values (2,1,2,1,2,12.45); insert into results_zabeg
values (3,1,4,4,3,14.21); insert into results_zabeg values
(4,1,3,3,4,15.23); insert into results_zabeg values
(5,2,1,2,3,16.23); insert into results_zabeg values
(6,2,2,1,1,12.43); insert into results_zabeg values
(7,2,3,3,2,12.56); insert into results_zabeg values
(8,3,1,1,1,12.23); insert into results_zabeg values
(9,3,2,2,2,14.23); insert into results_zabeg values
(10,3,4,3,3,15.03)
```

42.

43. Описание программного продукта.

44. Опишем процедуры и функции, которые вошли в пакет PACZABEG, созданный на языке SQL для работы с базой данных Скачки.

45. Функция Kolvo_Forces_Of_Vladel(Arg Forces.vladel_id%TYPE) возвращает натуральное значение, которое показывает количество лошадей, имеющих у заданного владельца (Arg – идентификатор владельца).

46. Процедура Forces_Of_Vladel(Arg vladel.vladel_name%TYPE) выводит список кличек лошадей, имеющих у заданного владельца (Arg – имя владельца). В процедуре используется курсор, содержащий запрос с выборкой, имеющий следующий вид:

```
47. Cursor CurForcel IS SELECT Forces.Force_name FROM forces WHERE
vladel_id=(select vladel_id from vladel where vladel_name=arg)
```

48. Процедура Inform_About_Vladel(Arg Forces.Force_Name%TYPE) выводит информацию о владельце (имя, адрес, телефон) по заданной кличке лошади (Arg). В процедуре используется курсор, со держащий запрос с выборкой, имеющий следующий вид:

```
49. Cursor CurVladel1 IS SELECT vladel_name,
vladel_address,vladel_telephon FROM Vladel WHERE vladel_id=(Select
vladel_id From Forces WHERE force_name=Arg)
```

50. Процедура Inform_About_Zabegs(Arg Zabeg.zabeg_data%TYPE) выводит информацию о забеге (клички участвующих лошадей, имена жокеев, занятые места и показанное в забеге время) по заданной дате забега (Arg). В процедуре используется курсор, содержащий запрос с выборкой, имеющий следующий вид:

```
51. Cursor CurZabeg1 IS SELECT Forces.force_name,Gokey.gokey_name,
    rzults_zabeg.rezult,results_zabeg.rez_time
52. FROM Fores,Gokey,results_zabeg
53.WHERE (Fores.force_id=results_zabeg.force_id)AND
    (Gokey.gokey_id=results_zabeg.gokey_id)AND
    results_zabeg.zabeg_id=(select zabeg_id from zabeg where
    zabeg_data=arg)
```

54. Процедура Lidere_Zabeg показывает лидеров всех забегов. В процедуре используется курсор, содержащий запрос с выборкой, имеющий следующий вид:

```
55. Cursor CurLider1 IS SELECT Forces.Force_name,Gokey.gokey_name,
    results_zabeg.rez_time,zabeg.zabeg_data
56. FROM Forces,Gokey,results_zabeg,zabeg
57. WHERE (results_zabeg.rezult=1)AND (forc-
    es.force_id=results_zabeg.force_id)AND
    (gokey.gokey_id=results_zabeg.gokey_id)AND
    (zabeg.zabeg_id=results_zabeg.zabeg_id)
```

58. Процедура New_Zabeg(Arg1 zabeg.zabeg_id%TYPE, Arg2 zabeg.zabeg_data%TYPE, Arg3 zabeg.zabeg_time%TYPE, Arg4 zabeg.ippodrom_id%TYPE, Arg5 zabeg.zabeg_name%TYPE, Arg6 results_zabeg.rez_id%TYPE, Arg7 results_zabeg.force_id%TYPE, Arg8 results_zabeg.gokey_id%TYPE, Arg9 results_zabeg.rezult%TYPE, Arg10 results_zabeg.rez_time%TYPE) осуществляет ввод данных для нового забега.

59. При этом происходит добавление в таблицах Zabeg и Rezults_Zabeg.

60. Создадим триггеры для обеспечения каскадных изменений в связанных таблицах базы данных Скачки.

61. • Триггер TRIG_Forces_Pol используется для таблицы Pol и обеспечивает каскадные изменения в связанных с ней таблицах.

62. • Триггер TRIG_Forces_Vladel используется для таблицы Vladel и обеспечивает каскадные изменения в связанных с ней таблицах.

63. • Триггер TRIG_Zabeg_Ipp используется для таблицы Ippodrom и обеспечивает каскадные изменения в связанных с ней таблицах.

64. • Триггер TRIG_RezZabeg_Force используется для таблицы Forces и обеспечивает каскадные изменения в связанных с ней таблицах.

65. • Триггер TRIG_RezZabeg_Gokey используется для таблицы Gokey и обеспечивает каскадные изменения в связанных с ней таблицах.

66. **Создание пакета.**

67. *Создание заголовка пакета:*

```
68. CREATE OR REPLACE PACKAGE PACZABEG AS FUNCTION
    Kolvo_Forces_Of_Vladel(Arg Forces.vladel_id%TYPE) RETURN INTEGER;
    PROCEDURE Forces_Of_Vladel (Arg vladel.vladel_name%TYPE); PROCEDURE
    Inform_About_Vladel (Arg Forces.Force_Name%TYPE); PROCEDURE In-
    form_About_Zabegs (Arg Zabeg.zabeg_data%TYPE); PROCEDURE
    Lidere_Zabeg; PROCEDURE New_Zabeg (Arg1 zabeg.zabeg_id%TYPE, Arg2
    zabeg.zabeg_data%TYPE, Arg3 zabeg.zabeg_time%TYPE, Arg4
    zabeg.ippodrom_id%TYPE, Arg6 results_zabeg.rez_id%TYPE, Arg7
```

```
results_zabeg.force_id%TYPE, Arg8 results_zabeg.gokey_id%TYPE, Arg9
results_zabeg.rezult%TYPE, Arg10 results_zabeg.rez_time%TYPE); end;
```

69. *Создание тела пакета:*

```
70.CREATE OR REPLACE PACKAGE BODY PACZABEG AS FUNCTION
Kolvo_Forces_Of_Vladel (Arg Forces.vladel_id%TYPE) RETURN INTEGER IS
n INTEGER; BEGIN Select COUNT(*) INTO n From Forces Where
vladel_id=Arg; RETURN n; END; PROCEDURE Forces_Of_Vladel (Arg
Vladel.vladel_name%TYPE) IS T1 Forces.Force_Name%TYPE; Cursor
CurForcel IS SELECT Forces.Force_name FROM forces WHERE
vladel_id=(select vladel_id from vladel where vladel_name=arg);
BEGIN Open CurForcel; FETCH CurForcel INTO T1; WHILE CurForcel%FOUND
LOOP DBMS_OUTPUT.PUT_LINE(CurForcel%ROWCOUNT||T1); FETCH CurForcel
INTO T1; END LOOP; Close CurForcel; END; PROCEDURE In-
form_About_Vladel (Arg Forces.Force_Name%TYPE) IS T1
Vladel.vladel_name%TYPE; T2 Vladel.vladel_address%TYPE; T3
Vladel.vladel_telephon%TYPE; Cursor CurVladell IS SELECT
vladel_name,vladel_address,vladel_telephon FROM Vladel WHERE
vladel_id=(Select vladel_id From Forces WHERE force_name=Arg); BEGIN
Open CurVladell; FETCH CurVladell INTO T1,T2,T3; WHILE
CurVladell%FOUND LOOP DBMS_OUTPUT.PUT_LINE(CurVladell%ROWCOUNT
||'-Name'||T1||'-Address '||T2||'-Telephon'||T3); FETCH CurVladell INTO
T1,T2,T3; END LOOP; Close CurVladell; END; PROCEDURE In-
form_About_Zabegs (Arg Zabeg.zabeg_data%TYPE) IS T1 Forc-
es.force_name%TYPE; T2 Gokey.gokey_name%TYPE; T3
results_zabeg.rezult%TYPE; T4 results_zabeg.rez_time%TYPE; Cursor
CurZabeg1 IS SELECT Forces.force_name,Gokey.gokey_name,
results_zabeg.rezult,results_zabeg.rez_time FROM Forc-
es,Gokey,results_zabeg WHERE (Forc-
es.force_id=results_zabeg.force_id)AND
(Gokey.gokey_id=results_zabeg.gokey_id)AND
results_zabeg.zabeg_id=(select zabeg_id from zabeg where
zabeg_data=arg); BEGIN Open CurZabeg1; FETCH CurZabeg1 INTO
T1,T2,T3,T4; WHILE CurZabeg1%FOUND LOOP 172 ♦ Глава 3. Задачи по
разработке простых баз данных 3.2. Пример выполнения лабораторной
работы DBMS_OUTPUT.PUT_LINE(CurZabeg1%ROWCOUNT
||'-ForceName
'||T1||'-Gokey ' ||T2||'-Mesto'||T3||'-Time'||T4); FETCH CurZabeg1
INTO T1,T2,T3,T4; END LOOP; Close CurZabeg1; END; PROCEDURE
Liders_Zabeg IS T1 Forces.Force_name%Type; T2 Gokey.gokey_name%TYPE;
T3 results_zabeg.rez_time%Type; T4 zabeg.zabeg_data%Type; Cursor
CurLider1 IS Select Forces.Force_name,Gokey.gokey_name,
results_zabeg.rez_time,zabeg.zabeg_data FROM Forc-
es,Gokey,results_zabeg,zabeg WHERE (results_zabeg.rezult=1)AND forc-
es.force_id=results_zabeg.force_id)AND
(gokey.gokey_id=results_zabeg.gokey_id)AND
(zabeg.zabeg_id=results_zabeg.zabeg_id); BEGIN Open CurLider1; FETCH
CurLider1 INTO T1,T2,T3,T4; WHILE CurLider1%FOUND LOOP
DBMS_OUTPUT.PUT_LINE(CurLider1%ROWCOUNT
||'-'||T1||'-'||T2||'-
'||T3||'-'||T4); FETCH CurLider1 INTO T1,T2,T3,T4; END LOOP; Close
CurLider1; END; PROCEDURE New_Zabeg (Arg1 zabeg.zabeg_id%TYPE, Arg2
zabeg.zabeg_data%TYPE, Arg3 zabeg.zabeg_time%TYPE, Arg4
zabeg.ippodrom_id%TYPE, Arg5 zabeg.zabeg_name%TYPE, Arg6
results_zabeg.rez_id%TYPE, Arg7 results_zabeg.force_id%TYPE, Arg8
```



```
results_zabeg.gokey_id%TYPE, Arg9 results_zabeg.rezult%TYPE, Arg10
results_zabeg.rez_time%TYPE ) IS BEGIN INSERT INTO zabeg
VALUES(arg1,arg2,arg3,arg4,arg5); COMMIT; INSERT INTO results_zabeg
VALUES(arg6,arg1,arg7,arg8,arg9,arg10); COMMIT; END; END; /
```

71. Создание триггеров:

```
72.CREATE OR REPLACE TRIGGER TRIG_Forces_Pol BEFORE UPDATE ON Pol FOR
EACH ROW BEGIN IF (:old.force_pol<>:new.force_pol) THEN UPDATE Forc-
es SET force_pol=:new.force_pol WHERE force_pol=:old.force_pol; END
IF; END; CREATE OR REPLACE TRIGGER TRIG_Forces_Vladel BEFORE UPDATE
ON vladel FOR EACH ROW BEGIN IF (:old.vladel_id<>:new.vladel_id)
THEN UPDATE Forces SET vladel_id=:new.vladel_id WHERE
vladel_id=:old.vladel_id; END IF; END; CREATE OR REPLACE TRIGGER
TRIG_Zabeg_Ipp BEFORE UPDATE ON Ippodrom FOR EACH ROW BEGIN
```

```
73. IF (:old.ippodrom_id<>:new.ippodrom_id) THEN UPDATE zabeg SET
ippodrom_id=:new.ippodrom_id WHERE ippodrom_id=:old.ippodrom_id; END
IF; END; CREATE OR REPLACE TRIGGER TRIG_RezZabeg_Force BEFORE UPDATE
ON Forces FOR EACH ROW BEGIN IF (:old.force_id<>:new.force_id) THEN
UPDATE results_zabeg SET force_id=:new.force_id WHERE
force_id=:old.force_id; END IF; END; CREATE OR REPLACE TRIGGER
TRIG_RezZabeg_Gokey BEFORE UPDATE ON gokey FOR EACH ROW BEGIN IF
(:old.gokey_id<>:new.gokey_id) THEN UPDATE results_zabeg SET
gokey_id=:new.gokey_id WHERE gokey_id=:old.gokey_id; END IF; END; /
```

6.4. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и/или опыта деятельности, характеризующие этапы формирования компетенций

Рейтинг по дисциплине «Технологии баз данных» (5 семестр)

Максимальная сумма баллов – 100.

Промежуточная аттестация – 70 баллов, зачет – 30 баллов.

Вид работы	Количество единиц работы	Количество баллов на единицу вида работы	Максимальная сумма баллов по виду работы
Посещение занятий в период семестра	42	0,5	21
Выполнение заданий для самостоятельной работы в течение выполнения лабораторных работ	12	2	24
Отчёт по индивидуальным заданиям	1	10	10
Решение задач повышенной сложности	2	5	10
Контрольная работа	1	5	5
Зачет	1	30	30

Рейтинг по дисциплине «Технологии баз данных» (6 семестр)

Составляющие итоговой оценки за дисциплину:

1) Текущий контроль (общий вес 60 баллов):

до 21 балла – посещение занятий;

до 26 баллов – выполнение заданий в ходе выполнения лабораторных работ в LMS Moodle и заданий для самостоятельной работы

до 10 баллов – выполнение отдельно выделенных в методических указаниях к выполнению проектного задания

до 3 баллов – выполнение контрольной работы в аудитории

2) Итоговый контроль заключается в проведении экзамена (общий вес - 40 баллов). Экзамен проводится по вопросам с обязательным решением задач. Как правило, студент получает два вопроса из приведенного выше списка и одну задачу на составление запроса, готовится в присутствии преподавателя и дает подробные комментарии. Студент, пропускавший занятия в ходе семестра, получает дополнительные вопросы и задачи по каждой пропущенной им теме (на усмотрение преподавателя).

Для получения положительной итоговой оценки на экзамене необходимо получить не менее 50% по каждой составляющей и выполнить все лабораторные работы. Шкала перевода баллов в оценку приведена в таблице.

Оценка	«отлично»	«хорошо»	«удовлетворительно»	«не удовлетворительно»
Интервал количества баллов	81..100	61..80	41..60	0..40

готовность к использованию основных моделей информационных технологий и способов их применения для решения задач в предметных областях (ПК 3)

Планируемые результаты обучения	Критерии оценивания с весовым коэффициентом	Показатели оценивания				
		1	2	3	4	5
Выпускник знает основные понятия в области проектирования баз данных	когнитивный – 0,1	Знает определения основных понятий в области теории баз данных, может приводить примеры	Знает о принципах и взаимосвязи между изученными понятиями	Знает области и особенности применения объектов, составляющих основу теории БД и СУБД	Знает характер процессов, происходящих в среде баз данных, связанные с основными понятиями	Знает тонкости процессов, происходящих в среде баз данных, связанные с основными понятиями
Выпускник знает основные технологии проектирования баз данных;	когнитивный – 0,1	Может перечислить виды технологий и сферы их применения	Правильно проектирует ввод данных различных стандартных типов; знает возможности языка для описания и реализации запросов к БД	Знает о способах описания данных составных и производных типов	Знает о проектировании данных с учетом соблюдения структуры и целостности	Знает тонкости проектирования и реализации запросов к данным
Выпускник знает этапы жизненного	когнитивный – 0,1	Знает основные этапы жизненного цикла баз	Знает и может аргументировать особенности каждого	Знает последовательность реализации и содержание	Знает способы реализации этапов жизненного	Осуществляет реализацию этапов жизненного цикла

Технологии баз данных				Б1.В.ОД.3		
цикла баз данных;		данных	го этапа жизненного цикла баз данных	основных этапов жизненного цикла баз данных	цикла баз данных	баз данных
Выпускник знает виды и способы организации запросов к данным в реляционных моделях	когнитивный – 0,1	Может перечислить виды и способы организации запросов к данным в реляционных моделях	Правильно проектирует виды и способы организации запросов к данным в реляционных моделях	Знает о способах организации запросов к данным в реляционных моделях	Знает о проектировании способов организации запросов к данным в реляционных моделях	Знает тонкости способов организации запросов к данным в реляционных моделях
Выпускник умеет выбирать технологию для проектирования реляционной модели данных;	деятельностный – 0,1	Способен понять правильно написанные коды запросов по типовым задачам	Способен понять и объяснить основные этапы проектирования программных кодов запросов к данным для решения типовых задач	Способен модифицировать коды запросов к данным по типовым задачам	Способен самостоятельно написать коды запросов к данным по типовым задачам	Способен самостоятельно написать, протестировать и исправить ошибки в кодах запросов к данным по типовым задачам
Выпускник владеет типовыми технологиями проектирования баз данных	деятельностный – 0,2	Не владеет типовыми технологиями проектирования баз данных	Имеет слабое представление об использовании типовых технологий проектирования баз данных	При рассмотрении правильно работающих кодов запросов к данным затрудняется в выделении этапов выполнения запросов	При написании кодов запросов к данным может выделить независимую подзадачу и обосновать необходимость ее оформления в виде подзапроса	При программировании кодов запросов к данным всегда выделяет порядок и приоритет операций, анализирует на семантическую и логическую истинность коды запросов к данным

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

7.1 Основная литература:

1. Медведкова И. Е. , Бугаев Ю. В. , Чикунов С. В. Базы данных: Воронеж: Воронежский государственный университет инженерных технологий, 2014. Доступно по ссылке <http://biblioclub.ru/index.php?page=book&id=336039&sr=1>
2. Гушин А. Н. Базы данных: учебно-методическое пособие. М., Берлин: Директ-Медиа, 2015. 311 с. Доступно по ссылке: <http://biblioclub.ru/index.php?page=book&id=278093&sr=1>

7.2 Дополнительная литература:

3. Базы данных: основы, проектирование, использование [Текст] : учебное пособие для студентов вузов / М. П. Малыгина. - 2-е изд. - СПб : БХВ-Петербург, 2012. - 528 с. : ил. - ISBN 9785941579419

4. Базы данных и Delphi [Текст] : теория и практика / Д. Осипов. - СПб. : БХВ-Петербург, 2011. - 752 с. - ISBN 978-5-9775-0659-5 : Б. ц. Доступ по ссылке: <http://ibooks.ru/reading.php?productid=22672>
5. Информатика: Базовый курс [Текст] : учебник / ред. С. В. Симонович. - 3-е изд. - СПб. : Питер, 2012. - 637 с. - ISBN 978-5-459-00439-7 :Б. ц. Доступ по ссылке: <http://ibooks.ru/reading.php?productid=23132>
6. Информационные системы и базы данных [Текст] : учебное пособие / В. Ю. Пирогов. - СПб. : БХВ-Петербург, 2013. - 528 с. - ISBN 978-5-9775-0399-0 : Б. ц. Доступ по ссылке: <http://ibooks.ru/reading.php?productid=18485>
7. Базы данных: модели и языки : [учеб. пособие для вузов по спец. и на- правлению "Прикл. математика и информатика" и по направлению "Информ. технологии"] / С.Д. Кузнецов. — Москва : Бином, 2008
8. Введение в реляционные базы данных : [учебное пособие для студентов высших учебных заведений, изучающих дисциплины "Базы данных", "Информ. системы"] / Владимир Кириллов, Геннадий Громов .— Санкт-Петербург : БХВ-Петербург, 2009 .— X, 454 с. : ил., табл.
9. Объектно-ориентированный анализ и проектирование с примерами приложений/ Гради Буч, Роберт А. Максимчук, Майкл У. Энгл, Бобби Дж. Янг, Джим Коаллен, Келли А. Хьюстон , 3-е издание, Вильямс, 720 стр., 2010
10. Oracle Database 10g XE для Windows : эффективное использование / Стив Бобровский .— Москва : ЛОРИ, 2009 .— XXVI, 486 с. : ил.

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

1. [_www.intuit.ru](http://www.intuit.ru)
2. [_http://microsoft.com/](http://microsoft.com/)
3. [_http://moodle.tsput.ru/course/view.php?id=11339](http://moodle.tsput.ru/course/view.php?id=11339)
4. ИКТ [Электронный ресурс] : федеральный образовательный портал / ФГАУ ГНИИ ИТТ "Информатика". - М. : [б. и.], 2003. - Загл. с титул. экрана. URL: <http://www.ict.edu.ru>

Курс обеспечен электронными учебными материалами, размещенными на сервере кафедры. Несмотря на общее единообразие разработки систем класса баз данных (БД), имеются вполне определенные особенности работы с каждой инструментальной системой (средой) создания БД; без практического навыка работы создание конкретной БД может вызвать излишние затруднения.

Рассматривается практика создания баз данных с использованием связки систем MySQL и языка скриптового программирования PHP.

Для качественной организации занятий в соответствии с разработанным курсом необходимо иметь возможность:

- 1) чтения лекций, в том числе с использованием мультимедийных средств обучения;
- 2) работы в компьютерном классе. На компьютерах должно быть установлено программное обеспечение:
 - a. утилита Putty;
 - b. текстовый редактор MicroSoftWord
- 3) выход студентов в образовательную систему MOODLE
- 4) доступ к личным страницам студентов, размещенным на кафедральном сервере.

9. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

Дисциплина «Проектирование баз данных» направлена на формирование у студентов знаний, умений и владений в области разработки и программирования баз данных. В результате изучения данной дисциплины студенты должны научиться решать задачи с помощью ЭВМ, раскладывать сложные задачи на независимые подзадачи, представлять, как можно использовать возможности компьютера для решения прикладных задач. Преподавателю необходимо провести систематизацию и выравнивание знаний студентов в области программирования на скриптовых языках, поскольку они могут сильно варьироваться вследствие различного начального уровня подготовки студентов.

Преподавание дисциплины должно включать в себя следующие образовательные технологии:

- 1) Организация лекций с использованием при необходимости мультимедийных технологий;
- 2) Использование в ходе лабораторных работ дидактических материалов в виде: опорных конспектов по теоретической составляющей занятий, файлов с примерами программ и т.п.
- 3) Использование ресурсов LMS MOODLE с целью организации процесса систематизации, приобретения и контроля знаний;
- 4) Организация выполнения индивидуальных заданий с целью усиления приобретения навыков самостоятельного построения моделей, разработки алгоритмов и написания программ;
- 5) Организация презентации индивидуальных заданий с целью формирования навыка владения профессиональной терминологией, грамотного выстраивания цепочки высказываний для доказательства обоснованности построенных моделей предметной области и правильности отражения реальных свойств объектов;
- 6) Формирование у студентов убежденности в необходимости последовательного освоения следующих этапов в образовательной деятельности:
 - a. ознакомься с содержанием и теоретическими основами изучаемой темы;
 - b. рассмотри, обсуди с другом и протестируй задачу, решенную кем-то;
 - c. реши самостоятельно задачу, подобную рассмотренной ранее;
 - d. реши самостоятельно задачу по изучаемой теме.

10. ПЕРЕЧЕНЬ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, ИСПОЛЬЗУЕМЫХ ПРИ ОСУЩЕСТВЛЕНИИ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ, ВКЛЮЧАЯ ПЕРЕЧЕНЬ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ И ИНФОРМАЦИОННЫХ СПРАВОЧНЫХ СИСТЕМ

Операционная система Windows.

Пакет прикладных программ MS Office.

Веб-браузеры.

Оболочка PhpMyAdmin.

Доступ студентов через личные кабинеты к электронным библиотечным системам.

Возможность работы студентов на удаленном рабочем столе кафедры информатики и информационных технологий.

11. ОПИСАНИЕ МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЙ БАЗЫ, НЕОБХОДИМОЙ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ «ТЕХНОЛОГИИ БАЗ ДАННЫХ»

Компьютерные аудитории, подключение к локальной сети университета и сети Интернет, права доступа к перечисленному выше программному обеспечению.

12. АННОТАЦИЯ РАБОЧЕЙ ПРОГРАММЫ ДИСЦИПЛИНЫ «ТЕХНОЛОГИИ БАЗ ДАННЫХ»

1. Планируемые результаты обучения по дисциплине, соотнесенных с планируемыми результатами освоения образовательной программы

Компетенция: *готовность к использованию современных системных программных средств: операционных систем, операционных и сетевых оболочек, сервисных программ (ПК 3).*

Выпускник знает:

- основные понятия в области теории баз данных;
- основные технологии обработки баз данных;
- этапы жизненного цикла баз данных;
- виды и способы организации запросов к данным в реляционных моделях;
- интерфейс и технологию работы с оболочкой PhpMyAdmin.

Умеет:

- выбирать технологию для проектирования реляционной модели данных;
- разрабатывать структуру модели данных;
- реализовывать этапы жизненного цикла баз данных;
- использовать технологии баз данных для обработки табличных данных;
- представлять отчеты по результатам обработки данных средствами СУБД;
- применять полученные знания при решении практических задач профессиональной деятельности;
- осуществлять поиск необходимой информации в глобальных компьютерных сетях и LMS MOODLE;

Владеет и (или) имеет опыт деятельности:

- умением работать в команде;
- владением методами проектирования и производства программного продукта, принципами построения, структуры и приемов работы с инструментальными средствами, поддерживая создание программного обеспечения;
- использованием основных моделей информационных технологий и способов их применения для решения задач в предметных областях;
- использованием классических приемов (алгоритмов) при решении корректных постановок классических задач;
- этапы проектирования и разработки баз данных;
- типовыми алгоритмами автоматизированной обработки баз данных.

2. Место дисциплины «Проектирование баз данных» в структуре ООП

Дисциплина «Технологии баз данных» относится к дисциплинам вариативной части обязательного профессионального цикла дисциплин направления. Изучение данной дисциплины базируется на освоении студентами дисциплин базовой части «Информатика и информационные технологии», «Вводный курс программирования», «Языки программирования», вариативной части профессионального цикла «Основы программирования».

К началу изучения дисциплины студенты должны владеть знаниями в области представления и моделирования данных, видов и назначения информационных технологий, умениями обрабатывать табличную информацию средствами специализированных информационных технологий и составлять программные коды на языках программирования объектно-ориентированной парадигмы.

Освоение данной дисциплины необходимо для качественного изучения дисциплин ООП, базирующихся на разработке и использовании технологии разработки баз данных, для выработки умений поиска, анализа и обработки данных в реляционных моделях, для квалифицированного ис-

пользования баз данных в профессиональной деятельности, для реализации программной разработки, которой посвящена выпускная квалификационная работа.

3. Объем дисциплины 7 (3+4) зачетных единиц.

4. Образовательный процесс осуществляется на русском языке.

5. Разработчики:

Ваныкина Галина Владиславовна, кандидат педагогических наук, доцент кафедры информатики и информационных технологий.

13. ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ К РАБОЧЕЙ ПРОГРАММЕ ДИСЦИПЛИНЫ «ТЕХНОЛОГИИ БАЗ ДАННЫХ»

Изменения к рабочей программе дисциплины отсутствуют.

Заведующий кафедрой

информатики и информационных технологий



А.В. Якушин,

«24» декабря 2015 г.

Программа составлена в соответствии с требованиями ФГОС ВО.

Разработчики:

Фамилия, имя, отчество	Учёная степень	Учёное звание	Должность	Дата разработки	Подпись
Ваныкина Галина Владиславовна	кандидат педагогических наук,	доцент	доцент кафедры информатики и информационных технологий	17.11.2015	