

МИНПРОСВЕЩЕНИЯ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования  
"Тульский государственный педагогический университет им. Л.Н. Толстого"  
(ФГБОУ ВО "ТГПУ им. Л.Н. Толстого")

**МОДУЛЬ "КОМПЬЮТЕРНЫЕ НАУКИ"**  
**Базы данных**

**рабочая программа дисциплины (модуля)**

Закреплена за кафедрой	<b>институт передовых информационных технологий</b>
ОПОП	<b>Направление 02.03.03 Математическое обеспечение и администрирование информационных систем направленность (профиль) Технологии программирования и анализ данных</b>
Квалификация	<b>Бакалавр</b>
Год начала подготовки	<b>2023</b>
Форма обучения	<b>очная</b>
Общая трудоемкость	<b>7 з.е.</b>

Виды контроля по семестрам:

экзамен 4  
зачет 3

Семестр(Курс.Номер семестра на курсе)	3(2.1)		4(2.2)		Итого	
	УП	РПД	УП	РПД	УП	РПД
Лекции	18	34	18	0	36	34
Лабораторные	34	34	34	0	68	34
Итого ауд.	52	68	52	0	104	68
КСР	2	4	2	0	4	4
Контактная работа	54	72	54	0	108	72
Сам. работа	90	72	18	0	108	72
Часы на контроль	0	0	36	36	36	36
Практическая подготовка	0	0	0	0	0	0
Семинары	0	0	0	0	0	0
Консультации	0	0	0	0	0	0
Итого трудоемкость в часах	144	144	108	36	252	180

Программу составил(и):

*д.п.н., зав. кафедрой, Богатырева Юлия Игоревна*

Рабочая программа дисциплины

**Базы данных**

разработана в соответствии с ФГОС:

Федеральный государственный образовательный стандарт высшего образования - бакалавриат по направлению подготовки 02.03.03 Математическое обеспечение и администрирование информационных систем (приказ Минобрнауки России от 23.08.2017 г. № 809)

составлена на основании учебного плана:

Направление 02.03.03 Математическое обеспечение и администрирование информационных систем  
направленность (профиль) Технологии программирования и анализ данных  
утвержденного Учёным советом вуза от 27.10.2022 протокол № 13.

РПД утверждена Учёным советом университета  
протокол от 27.10.2022 г. № 13

**1. ЦЕЛИ ОСВОЕНИЯ ДИСЦИПЛИНЫ**

Выработка практических навыков проектирования, разработки и использования баз данных

**2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ООП**

Цикл (раздел) ООП:	Б1.О.06
<b>2.1</b>	<b>Требования к предварительной подготовке обучающегося:</b>
1.	К началу изучения дисциплины студенты должны владеть знаниями в области представления и моделирования данных, этапов жизненного цикла баз данных, умениями обрабатывать табличную информацию средствами специализированных информационных технологий, владеть основными технологиями разработки баз данных.
2.	Деловая коммуникация и основы деловой этики
<b>2.2</b>	<b>Дисциплины и практики, для которых освоение данной дисциплины (модуля) необходимо как предшествующее:</b>
1.	Освоение данной дисциплины необходимо для качественного изучения дисциплин ОПОП, базирующихся на разработке и программировании баз данных, для выработки умений аналитической обработки данных автоматизированными средствами, для квалифицированного использования баз данных в профессиональной деятельности, для реализации программной разработки, которой посвящена выпускная квалификационная работа
2.	Экономические информационные системы
3.	Практикум по программированию мобильных приложений
4.	Средства разработки кроссплатформенного программного обеспечения
5.	Разработка экономических информационных систем

**3. СООТНЕСЕНИЕ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ (МОДУЛЮ) С ИНДИКАТОРАМИ ДОСТИЖЕНИЯ КОМПЕТЕНЦИЙ****3.1 Компетенции обучающегося и индикаторы их достижения:**

ОПК-5: Способен устанавливать и сопровождать программное обеспечение для информационных систем и баз данных, в том числе отечественного производства

ОПК-5.1	Знает методику установки и администрирования информационных систем и баз данных основные технологии обработки баз данных; этапы жизненного цикла баз данных;
ОПК-5.2	Умеет реализовывать техническое сопровождение информационных систем и баз данных реализовывать этапы жизненного цикла баз данных;
ОПК-5.3	Имеет практические навыки установки и инсталляции программных комплексов владения методами проектирования и производства программного продукта, принципами построения, структуры и приемов работы с инструментальными средствами, поддерживая создание программного обеспечения;

ПК-3: Способен разрабатывать политику информационной безопасности и регламенты систем безопасности в информационных системах и базах данных

ПК-3.1	Знает проблемы угроз безопасности, способы их предотвращения и методики разработки регламента аудита систем безопасности на уровне обработки данных. виды и способы организации запросов к данным в реляционных моделях.
ПК-3.2	Умеет использовать знания по выявлению угроз безопасности на уровне обработки данных и оценивать степень их защиты разрабатывать структуру модели данных;
ПК-3.3	Владеет навыками выбора основных средств поддержки информационной безопасности на уровне обработки данных использования основных моделей информационных технологий и способов их применения для решения задач в предметных областях;

УК-3: Способен осуществлять социальное взаимодействие и реализовывать свою роль в команде;

УК-3.1	Демонстрирует способность работать в команде, проявляет лидерские качества и умения использовать технологии баз данных для обработки табличных данных.
--------	---

**3.2 Результаты обучения по дисциплине:****В результате освоения дисциплины обучающийся должен:**

	<b>Знать:</b>
3.1	основные технологии обработки баз данных;
3.2	этапы жизненного цикла баз данных;
3.3	виды и способы организации запросов к данным в реляционных моделях.

	<b>Уметь:</b>
У.1	разрабатывать структуру модели данных;
У.2	реализовывать этапы жизненного цикла баз данных;
У.3	использовать технологии баз данных для обработки табличных данных.
	<b>Владеть:</b>
В.1	владения методами проектирования и производства программного продукта, принципами построения, структуры и приемов работы с инструментальными средствами, поддерживая создание программного обеспечения;
В.2	использования основных моделей информационных технологий и способов их применения для решения задач в предметных областях;

#### 4. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

Код занятия	Наименование разделов и тем /вид занятия/	Семестр / Курс	Часов	Литература	Содержание
	<b>Тема 1. Основные понятия в области проектирования баз данных</b>				
1.1	Основные понятия в области проектирования баз данных /Лек/	3	4	Л1.1 Л1.2 Л1.3Л2.2	Банк данных: состав и основные понятия. СУБД: определение, классификация, состав. Трехуровневая архитектура базы данных: внешний, концептуальный и внутренний уровни. Понятие логической и физической независимости данных.
1.2	Анализ предметной области /Лаб/	3	4	Л1.1 Л1.2 Л1.3Л2.1 Л2.2 Л2.3	Предметная область. Объекты и атрибуты, связи между объектами и атрибутами объектов. Модель предметной области. Концептуальная модель. Типы моделей данных.
1.3	Основные понятия в области проектирования баз данных /Ср/	3	2	Л1.1 Л1.2 Л1.3Л2.2	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.
	<b>Тема 2. Методы проектирования и модели данных</b>				
2.1	Методы проектирования и модели данных /Лек/	3	4	Л1.1 Л1.2 Л1.3Л2.2	Иерархическая модель данных: определение, элементы данных. Сетевая модель данных: определение, элементы данных. Реляционная модель данных: определение, элементы данных. Операции реляционной алгебры: объединение, пересечение, разность, проекция, выборка, деление, декартово произведение, соединение. Типы взаимосвязей: «один – к – одному», «один – ко – многим», «многие – к – одному» и «многие – ко – многим». Контроль целостности связи.
2.2	Реляционная алгебра и отношения /Лаб/	3	4	Л1.1 Л1.2 Л1.3Л2.1 Л2.2 Л2.3	Выполнение операций над отношениями. Языки манипулирования данными. Основные операции реляционной алгебры: объединение, пересечение, разность, декартово произведение, проекция, частное и другие. Примеры. Реляционное исчисление с переменными-кортежами. Реляционное исчисление с переменными-доменами. Понятие безопасности реляционного исчисления. Сравнение алгебраических языков и языков исчисления. Дополнительные возможности языков манипулирования данными в реляционных системах.

2.3	Методы проектирования и модели данных /Ср/	3	6	Л1.1 Л1.2 Л1.3Л2.2	Цели проектирования. Универсальное отношение и проблемы его использования. Функциональные зависимости /ФЗ/. Декомпозиция отношения. Нормальная форма Бойса-Кодда /НФБК/. Избыточные ФЗ. Правила вывода.
	<b>Тема 3. Этапы проектирования баз данных</b>				
3.1	Этапы проектирования баз данных /Лек/	3	4	Л1.1 Л1.2 Л1.3Л2.2	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.
3.2	Построение модели "сущность-связь" /Лаб/	3	4	Л1.1 Л1.2 Л1.3Л2.1 Л2.2 Л2.3	Модель «сущность-связь» (ER - модель) и ее основные нотации. Правила перехода от ER - модели к реляционной модели. Основные этапы проектирования БД методом «сущность-связь».
3.3	Этапы проектирования баз данных /Ср/	3	8	Л1.1 Л1.2 Л1.3Л2.2	Проектирование структуры данных. Этапы проектирования баз данных. Онтологическое моделирование. Инфологическая (информационно-логическая) модель. Преобразование инфологической модели в даталогическую. Физическое проектирование
	<b>Тема 4. Метод нормальных форм</b>				
4.1	Метод нормальных форм /Лек/	3	4	Л1.2 Л1.3Л2.2	Концепция нормальных форм, аномалии удаления, добавления и изменения. Процесс нормализации отношения, способы нормализации. Взаимосвязи между атрибутами. Первая нормальная форма (1НФ), вторая нормальная форма (2 НФ), третья нормальная форма (3 НФ). Нормальные формы более высокого порядка. Многочленные зависимости и четвертая нормальная форма (4НФ). Зависимости соединения и пятая нормальная форма (5НФ). Другие нормальные формы.
4.2	Нормализация отношений /Лаб/	3	4	Л1.2 Л1.3Л2.1 Л2.2 Л2.3	Нормальные формы схем отношений. Первая нормальная форма. Вторая нормальная форма. Примеры. Третья нормальная форма. Примеры. Нормальная форма Бойса-Кодда. Примеры. Многочленные зависимости. Аксиомы многочленных зависимостей. Четвертая нормальная форма. Примеры.
4.3	Метод нормальных форм /Ср/	3	10	Л1.2 Л1.3Л2.2	Нормализация отношений. Примеры. Недостатки и преимущества процесса нормализации. Алгоритм нормализации ER-модели
	<b>Тема 5. Метод ER-диаграмм</b>				
5.1	Метод ER-диаграмм /Лек/	3	4	Л1.2 Л1.3Л2.2	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.

5.2	Графические средства построения ER-диаграмм /Лаб/	3	4	Л1.2 Л1.3Л2.1 Л2.2 Л2.3	Этап инфологического моделирования данных. Построение логической модели данных. Создать логические модели данных по выбранной предметной области.
5.3	Метод ER-диаграмм /Ср/	3	8	Л1.2 Л1.3Л2.2	Основные понятия модели "сущность-связь". Связи. Типы связей. Документирование отношений. Нотация Мартина, нотация IDEF1X, нотация Баркера и др. CASE-средства: Design/IDEF, Power Designer (S-Designer), Oracle, ERWin
	<b>Тема 6. Инструментальные средства проектирования структур данных</b>				
6.1	Инструментальные средства проектирования структур данных /Лек/	3	4	Л1.1 Л1.2 Л1.3Л2.2	Case-средства: назначение, основные возможности. Жизненный цикл информационной системы. ВРwin. 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. Создание функциональной модели на основе методологии IDEF0, DFD.
6.2	Имитационное моделирование /Лаб/	3	6	Л1.1 Л1.2 Л1.3Л2.1 Л2.2 Л2.3	ВРwin. 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.
6.3	Инструментальные средства проектирования структур данных /Ср/	3	8	Л1.1 Л1.2 Л1.3Л2.2	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. Создание функциональной модели на основе методологии IDEF0, DFD.
	<b>Тема 7. Операторы определения данных</b>				

7.1	Операторы определения данных /Лек/	3	4	Л1.2 Л1.3Л2.1 Л2.2 Л2.4	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.
7.2	PHP и MySQL /Лаб/	3	4	Л1.2 Л1.3Л2.1 Л2.2 Л2.3 Л2.4	Работа с сервером баз данных MySQL средствами языка PHP. Разные функции MySQL. Соединение с сервером MySQL. Соединение с базой данных. Функции PHP для взаимодействия с MySQL. Заккрытие соединения. Выполнение запросов. Типы таблиц. Создание баз данных и таблиц. Удаление базы данных и таблиц.
7.3	Операторы определения данных /Ср/	3	16	Л1.3Л2.2 Л2.4	Используйте разработанную главную страницу Вашего Приложения. Реализуйте переход с Главной страницы к страницам для просмотра таблиц базы данных. На основе шаблона Главной страницы создайте страницу сценария (расширение *.php), где выполните соединение с сервером баз данных MySQL
	<b>Тема 8. Операторы манипулирования и выборки данных</b>				
8.1	Операторы манипулирования и выборки данных /Лек/	3	6	Л1.2 Л1.3Л2.2 Л2.4	Полный и сокращенный синтаксис оператора добавления данных. Синтаксис операторов на обновление и удаление данных. Оператор выборки записей. Агрегатные функции. Группировка и упорядочение записей. Применение выражений и функций в операторе SELECT, классификация функций, агрегация данных, условные выражения.
8.2	Создание запросов MySQL /Лаб/	3	4	Л1.2 Л1.3Л2.1 Л2.2 Л2.3 Л2.4	Условный вариант оператора SELECT. Запросы. Функции агрегирования. Группировка. Соединение. Связанные запросы. Объединение.
8.3	Операторы манипулирования и выборки данных /КСР/	3	4	Л1.2 Л1.3Л2.2 Л2.4	Для получения данных для вставки данных в БД используйте интерфейс web-приложения, с помощью которого на странице предусмотрите поля для ввода значений и дальнейшей передачи их в сценарий. Для удаления данных предусмотрите ввод ключевых полей, которые идентифицируют записи таблицы. После выполнения операций вставки или удаления предусмотрите вывод сообщений для пользователя о выполнении операции. Например, «Запись добавлена в таблицу» или «Из таблицы удалена запись».
	<b>Подготовка к зачету</b>				
9.1	Подготовка к зачету /Ср/	3	14	Л1.2 Л1.3Л2.2	Вопросы к зачету

## 5. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ

### 5.1. Типовые задания для проведения текущего контроля

Примерный список заданий на проектирование модели данных в соответствии с особенностями предметной области  
Решение заданий, использующих объекты базы данных, предполагает выполнение следующих этапов:

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.
2. Проектирование схемы базы данных: описание схем таблиц, типов (доменов) атрибутов, определение ограничений целостности.

### 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 хранящейся 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) позволять 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 заносится 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 постройки. Фирма регистрирует каждый выход на лов, записывая название катера, имена и адреса членов команды с указанием их должностей (капитан, боцман и т.д.), даты выхода и возвращения, а также вес пойманной рыбы отдельно по сортам (например, трески). За время одного рейса катер может посетить несколько банок. Фиксируется дата прихода на каждую банку и дата отплытия, качество выловленной рыбы (отличное, хорошее, плохое). На борту улов 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, именуется 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, выставляются 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 выставлять любое количество вещей, а покупатель имеет право приобретать любое количество вещей. Одно и то же лицо или фирма может выступать и как продавец, и как покупатель. После аукциона служащие фирмы, проводящей аукционы, записывают фактическую цену, уплаченную за проданный предмет, и фиксируют данные покупателя. 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 спра-вочник 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 проце-дур и функций, позволяющий: 1) для указанной даты 1-ой рассылки вывести список приглашен-ных и посчитать их количество; 2) предоставить возможность добавления приглашенных на кон-ференцию с указанием оргвзноса и даты его уплаты; 3) вывести список приглашенных, с указани-ем даты об уплате оргвзноса; 4) для указанной интервала дат, вывести список участников, упла-тивших оргвзнос в этом диапазоне; 5) для указанного города вывести название тезисов докладов, поступивших из этого города; 6) для указанного города, вывести список нуждающихся в гостини-це. Предусмотреть разработку триггеров, обеспечивающих каскадные изменения в связанных таб-лицах.

11. База данных для обслуживания склада. База данных должна обеспечить автоматизацию складского учета. В ней должны содержаться следующие данные: информация о единицах хране-ния, 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 проце-дур 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 постав-щиков, решив самостоятельно наладить их производство. С этой целью она организовала сеть специализированных цехов, каждый из которых принимает определенное участие в технологиче-ском 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, но- мер служащего, ответственного за выдачу, дата и время выдачи, номер операции и номер наряда на проведение работ, который будет обсуждаться ниже. Реально затраченное количество мат риала 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 инстру-ментов этих типов, которые должны использоваться при ее выполнении. Когда инструменты дей-ствительно берутся со склада, фиксируется индивидуальный номер каждого экземпляра, указыва-ются номер заказавшего их цеха и номер наряда на проведение работ. И в этом случае затребован-ное 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 ин - формация.

Помимо SQL запросов для создания таблиц базы данных, раз- работать пакет, состоящий из про-цедур 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 расходующих материалов, используемых в раз- личных 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 каскад- ные изменения в связанных таблицах.

13. База данных музыкального магазина. Таблицы базы данных содержат информацию о музы-кантах, 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 про-цедур 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

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.

Требуется: • сформировать структуру таблиц базы данных; • подобрать подходящие имена табли-цам 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.

Описание структуры таблиц базы данных. Опишем структуру таблиц, связи между ними и произведем их нормализацию.

Создадим две вспомогательные таблицы Pol и Ippodrom.

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 и 2 женский

Таблица Ippodrom содержит возможные названия ипподромов (поле Ippodrom\_Name) и их поряд-ковые номера (поле Ippodrom\_id). Поле Ippodrom\_id является первичным ключом таблицы. Ippodrom\_id Ippodrom\_Name

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).

Таблица Gokey Force\_Age – содержит информацию о возрасте лошади. Vlade1\_id – содержит ин-формацию о владельце лошади и является внешним ключом (ссылается на поле Vlade1\_id таблицы Vlade1).

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 ≥ 18).

Gokey\_Reiting – содержит информацию о рейтинге жокея и имеет ограничение (Gokey\_Reiting ≥ 0).

Таблица Vlade1 содержит информацию о владельцах лошадей. Эта таблица содержит следующие поля. Vlade1\_id – содержит идентификатор владельца и это поле объ-явлено как первичный ключ. Vlade1\_Name – содержит имя владельца.

Vlade1\_Address – содержит адрес владельца. Vlade1\_Telephon – содержит телефон владельца.

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 – со-держит 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.

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 ин-формацию о занятом месте.

Rez\_Time содержит информацию о показанном в заезде времени.

Концептуальная модель базы данных приведена на Рисунке 1:

Рисунок 1 Концептуальная модель базы данных

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.

Создание и заполнение таблиц.

Таблица Пол:

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

Заполнение таблицы:

```
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'); insert into ippodrom values (2,'Sportsmen'); insert into ippodrom values (3,'Professional');
```

Таблица Лошади:

```
create table forces (force_id integer primary key, force_name var-char(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);
```

Заполнение таблицы:

```
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);
```

Таблица Жокеи:

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

Заполнение таблицы:

```
insert into gokey values (1,'Ivlev','Moskva',21,314); insert into gokey values (2,'Semenov','Moskva',35,245); insert into gokey values (3,'Nikitin','Voronezh',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);
```

Таблица Владельцы:

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

Заполнение таблицы:

```
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);
```

Таблица Забеги:

```
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);
```

Заполнение таблицы:

```
insert into zabeg values (1,'1-APR-02','18:30',1,'Aprelski'); insert into zabeg values (2,'3-MAY-02','12:30',3,'Mayski'); insert into 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,'Febrary'); insert into zabeg values (6,'15-APR-03','11:00',3,'Aprelski03'); insert into zabeg values (7,'1-OCT-03','18:30',1,'October03');
```

Таблица Результаты забегов:

```
create table rezults_zabeg (rez_id integer primary key, zabeg_id integer, force_id integer, gokey_id integer, rezult integer not null, rez_time real not null, FOREIGN KEY(force_id) REFERENCES forces, FOREIGN KEY(gokey_id) REFERENCES gokey, FOREIGN KEY(zabeg_id) REFERENCES zabeg);
```

Заполнение таблицы:

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

Описание программного продукта.

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

Функция Kolvo\_Forces\_Of\_Vladel(Arg Forces.vladel\_id%TYPE) возвращает натуральное значение, которое показывает количество лошадей, имеющих у заданного владельца (Arg – идентификатор владельца).

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:

```
Cursor CurForce1 IS SELECT Forces.Force_name FROM forces WHERE vladel_id=(select vladel_id from vladel where vladel_name=arg)
```

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, содержащий запрос с выборкой, имеющий следующий вид:

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

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:

```
Cursor CurZabeg1 IS SELECT Forces.force_name, Gokey.gokey_name, rezults_zabeg.rezult, rezults_zabeg.rez_time FROM Forces, Gokey, rezults_zabeg
```

```
WHERE (Fores.force_id=rezults_zabeg.force_id)AND (Gokey.gokey_id=rezults_zabeg.gokey_id)AND rezults_zabeg.zabeg_id=
(select zabeg_id from zabeg where zabeg_data=arg)
```

Процедура Lidere\_Zabeg показывает лидеров всех забегов. В процедуре используется курсор, со-держащий запрос с выборкой, имеющий следующий вид:

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

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 rezults\_zabeg.rez\_id%TYPE, Arg7 rezults\_zabeg.force\_id%TYPE, Arg8 rezults\_zabeg.gokey\_id%TYPE, Arg9 rezults\_zabeg.rezult%TYPE, Arg10 rezults\_zabeg.rez\_time%TYPE) осуществляет ввод данных для нового забега.

При этом происходит добавление в таблицах Zabeg и Rezults\_Zabeg.

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

- Триггер TRIG\_Forces\_Pol используется для таблицы Pol и обеспечивает каскадные изменения в связанных с ней таблицах.
- Триггер TRIG\_Forces\_Vladel используется для таблицы Vladel и обеспечивает каскадные изменения в связанных с ней таблицах.
- Триггер TRIG\_Zabeg\_Ipp используется для таблицы Ippodrom и обеспечивает каскадные изменения в связанных с ней таблицах.
- Триггер TRIG\_RezZabeg\_Force используется для таблицы Forces и обеспечивает каскадные изменения в связанных с ней таблицах.
- Триггер TRIG\_RezZabeg\_Gokey используется для таблицы Gokey и обеспечивает каскадные изменения в связанных с ней таблицах.

Создание пакета.

Создание заголовка пакета:

```
CREATE OR REPLACE PACKAGE PACZABEG AS FUNCTION Kol-vo_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 Inform_About_Zabegs (Arg Zabeg.zabeg_data%TYPE); PROCEDURE
Liders_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 rezults_zabeg.rez_id%TYPE, Arg7 rezults_zabeg.force_id%
TYPE, Arg8 rezults_zabeg.gokey_id%TYPE, Arg9 rezults_zabeg.rezult%TYPE, Arg10 rezults_zabeg.rez_time%TYPE); end;
```

Создание тела пакета:

```
CREATE OR REPLACE PACKAGE BODY PACZABEG AS FUNCTION Kol-vo_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 CurForce1
IS SELECT Forces.Force_name FROM forces WHERE vladel_id=(select vladel_id from vladel where vladel_name=arg); BEGIN
Open CurForce1; FETCH CurForce1 INTO T1; WHILE CurForce1%FOUND LOOP DBMS_OUTPUT.PUT_LINE(CurForce1%
ROWCOUNT||T1); FETCH CurForce1 INTO T1; END LOOP; Close CurForce1; END; PROCEDURE Inform_About_Vladel (Arg
Forces.Force_Name%TYPE) IS T1 Vladel.vladel_name%TYPE; T2 Vladel.vladel_address%TYPE; T3 Vladel.vladel_telephon%
TYPE; Cursor CurVladel1 IS SELECT vladel_name,vladel_address,vladel_telephon FROM Vladel WHERE vladel_id=(Select
vladel_id From Forces WHERE force_name=Arg); BEGIN Open CurVladel1; FETCH CurVladel1 INTO T1,T2,T3; WHILE
CurVladel1%FOUND LOOP DBMS_OUTPUT.PUT_LINE(CurVladel1%ROWCOUNT ||'-Name'||T1||'-Address' ||T2||'-
Telephon'||T3); FETCH CurVladel1 INTO T1,T2,T3; END LOOP; Close CurVladel1; END; PROCEDURE Inform_About_Zabegs
(Arg Zabeg.zabeg_data%TYPE) IS T1 Forces.force_name%TYPE; T2 Gokey.gokey_name%TYPE; T3 rezults_zabeg.rezult%
TYPE; T4 rezults_zabeg.rez_time%TYPE; Cursor CurZabeg1 IS SELECT Forc-es.force_name,Gokey.gokey_name, re-
zults_zabeg.rezult,rezults_zabeg.rez_time FROM Forc-es,Gokey,rezults_zabeg WHERE (Forc-es.force_id=rezults_zabeg.force_id)
AND (Gokey.gokey_id=rezults_zabeg.gokey_id)AND rezults_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 Lidere_Zabeg IS T1 Forces.Force_name%
Type; T2 Gokey.gokey_name%TYPE; T3 rezults_zabeg.rez_time%Type; T4 zabeg.zabeg_data%Type; Cursor CurLider1 IS Select
Forces.Force_name,Gokey.gokey_name, re-zults_zabeg.rez_time,zabeg.zabeg_data FROM Forc-es,Gokey,rezults_zabeg,zabeg
WHERE (rezults_zabeg.rezult=1)AND forc-es.force_id=rezults_zabeg.force_id)AND (gokey.gokey_id=rezults_zabeg.gokey_id)
AND (zabeg.zabeg_id=rezults_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 rezults_zabeg.rez_id%TYPE, Arg7 rezults_zabeg.force_id%TYPE, Arg8 rezults_zabeg.gokey_id%TYPE, Arg9
rezults_zabeg.rezult%TYPE, Arg10 rezults_zabeg.rez_time%TYPE ) IS BEGIN INSERT INTO zabeg VALUES
(arg1,arg2,arg3,arg4,arg5); COMMIT; INSERT INTO rezults_zabeg VALUES(arg6,arg1,arg7,arg8,arg9,arg10); COMMIT; END;
END; /
```

Создание триггеров:

```
CREATE OR REPLACE TRIGGER TRIG_Forces_Pol BEFORE UPDATE ON Pol FOR EACH ROW BEGIN IF
(:old.force_pol<>:new.force_pol) THEN UPDATE Forces 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
```

```
IF (:old.ippodrom_id<>:new.ippodrom_id) THEN UPDATE zabeg SET ippo-drom_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 rezults_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 rezults_zabeg SET gokey_id=:new.gokey_id WHERE gokey_id=:old.gokey_id; END IF; END; /
```

### 5.2. Типовые задания для проведения промежуточной аттестации

Примерный перечень вопросов к зачету

1. Методологические основы проектирования информационных систем
2. Основы технологии проектирования информационных систем
3. Планирование и контроль проектных работ
4. Каноническое проектирование информационной системы
5. Проектирование информационного обеспечения
6. Проектирование документальных баз данных
7. Проектирование фактографических баз данных
8. Основные понятия, архитектура и классификация CASE-средств
9. Сущность функционального (структурного) подхода
10. Методология функционального моделирования SADT
11. Моделирование потоков данных (процессов), DFD – диаграммы потоков данных;
12. Моделирование данных. Case-метод Баркера
13. Методология IDEF1
14. Проектирование ИС. Локальные CASE-средства (ERwin, BPwin)
15. Сущность объектно-ориентированного подхода к проектированию информационных систем
16. Унифицированный язык моделирования UML
17. Технологическая суть объектно-ориентированного проектирования ИС
18. Основные понятия и классификация методов типового проектирования
19. Классификация, примеры типовых информационных систем и их характеристика
20. Методы конфигурирования типовой информационной системы
21. Сущность параметрически-ориентированного проектирования ИС
22. Технологическая сеть проектирования с помощью параметрической настройки функционального пакета прикладных программ
23. Сущность модельно-ориентированного проектирования
24. Построение бизнес-модели предприятия
25. Технологическая сеть модельно-ориентированного проектирования ИС
26. Основные положения методологии RAD
27. Инструментальные средства для разработки приложений RAD
28. Варианты проектирования с использованием систем-прототипов
29. Реинжиниринг бизнес-процессов на основе интегрированных информационных систем
30. Принципы и особенности проектирования интегрированных информационных систем
31. Проектирование системы АРМ на основе локальной вычислительной сети
32. Системы управления информационными потоками как средство интеграции приложений
33. Проектирование интегрированных информационных систем
34. Проектирование клиент-серверных информационных систем
35. Проектирование систем оперативной обработки транзакций
36. Проектирование систем оперативного анализа данных.
37. Проектирование процессов защиты данных
38. Стандарты информационной безопасности
39. Проектирование системы защиты данных в информационных базах
40. Стандартные методы совместного доступа к базам и программам в сложных ИС

### 5.3. Перечень видов оценочных средств

1. Самостоятельная работа
2. Лабораторная работа
3. Зачёт

### 5.4. Процедура применения оценочных материалов

Промежуточная аттестация может проводиться с применением электронного обучения и (или) дистанционных образовательных технологий в соответствии с "Порядком проведения промежуточной аттестации с применением электронного обучения и /или дистанционных образовательных технологий".

Оценочные материалы по дисциплине приведены в приложении к РПД в файле "ПБД\_Ванькина.pdf"

Рейтинг по дисциплине «ПРОЕКТИРОВАНИЕ БАЗ ДАННЫХ»

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 зачета (общий вес - 40 баллов).

<b>6. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)</b>				
<b>6.1. Рекомендуемая литература</b>				
<b>6.1.1. Основная литература</b>				
	Авторы, составители	Заглавие	Издательство, год (кол-во экземпляров для печатных изданий)	Ссылка на электронное издание
Л1.1	Туманов В. Е.	Основы проектирования реляционных баз данных: учебное пособие	, 2007	<a href="http://biblioclub.ru/index.php?page=book&amp;id=233290">http://biblioclub.ru/index.php?page=book&amp;id=233290</a>
Л1.2	Гущин А. Н.	Базы данных: учебно-методическое пособие	, 2015	<a href="http://biblioclub.ru/index.php?page=book&amp;id=278093">http://biblioclub.ru/index.php?page=book&amp;id=278093</a>
Л1.3	Карпова Т. С.	Базы данных: модели, разработка, реализация : учебное пособие	, 2016	<a href="http://biblioclub.ru/index.php?page=book&amp;id=429003">http://biblioclub.ru/index.php?page=book&amp;id=429003</a>
<b>6.1.2. Дополнительная литература</b>				
	Авторы, составители	Заглавие	Издательство, год (кол-во экземпляров для печатных изданий)	Ссылка на электронное издание
Л2.1	Чубукова И. А.	Data Mining	, 2008	<a href="http://biblioclub.ru/index.php?page=book&amp;id=233055">http://biblioclub.ru/index.php?page=book&amp;id=233055</a>
Л2.2	Медведкова И. Е., Бугаев Ю. В., Чикунев С. В.	Базы данных	, 2014	<a href="http://biblioclub.ru/index.php?page=book&amp;id=336032">http://biblioclub.ru/index.php?page=book&amp;id=336032</a>
Л2.3	Сирант О. В., Коваленко Т. А.	Работа с базами данных	, 2016	<a href="http://biblioclub.ru/index.php?page=book&amp;id=428978">http://biblioclub.ru/index.php?page=book&amp;id=428978</a>
Л2.4	Кияев В., Граничин О.	Безопасность информационных систем: курс	Москва: Национальный Открытый Университет «ИНТУИТ», 2016	<a href="http://biblioclub.ru/index.php?page=book&amp;id=429032">http://biblioclub.ru/index.php?page=book&amp;id=429032</a>
<b>6.2. Перечень ресурсов информационно-телекоммуникационной сети "Интернет"</b>				
Э1	«ИНТУИТ» [Электронный ресурс] : национальный открытый университет / ООО «Интуит.ру». М. : [б. и.], 2016. - Загл. с титул. экрана. - Б. ц.			
Э2	ИКТ [Электронный ресурс] : федеральный образовательный портал / ФГАУ ГНИИ ИТТ "Информатика". - М. : [б. и.], 2003. - Загл. с титул. экрана.			
<b>6.3. Информационные технологии</b>				
<b>6.3.1 Перечень лицензионного и свободно распространяемого программного обеспечения</b>				
1.	Операционная система Microsoft Windows XP Professional Russian. Лицензия № 16698685 от 08.08.2003 г.			
2.	Операционная система Microsoft Windows Professional 7 Russian. Лицензия №48497058 от 13.05.2011 г., договор № Пр/16/6 от 05 апреля 2016 г.			
3.	Операционная система Microsoft Windows 10 Professional Russian. Контракт № ПР/ФЕН/15/18 от 23.10.2015 г., договор № Пр/16/6 от 05 апреля 2016 г.			
4.	Программное обеспечение Microsoft Office Enterprise 2007 Russian. Лицензия №46138962 от 16.11.2009			
5.	Программное обеспечение Microsoft Office 2013 Professional. Контракт № 405535 от 2 ноября 2015 года, контракт № ПР/ФЕН/15/18 от 23.10.2015 г.			
6.	Программа для распознавания текста ABBYY FineReader 9.0 Corporate Edition. Лицензионный сертификат - код позиции AF90-3U1V25-102, ABBYY FineReader 9.0 Corporate Edition Volume License Concurrent от 28 июля 2009 г.			
7.	Электронный словарь ABBYY Lingvo X3 Европейская версия - Код позиции AL14-2U1V05-102, ABBYY Lingvo x3 Европейская версия. Именная лицензия Concurrent от 28 июля 2009 г.			

8.	Комплексная система антивирусной защиты Kaspersky Endpoint Security для бизнеса – стандартный Russian Edition. 500-999 Node 2 year Educational Renewal License. Лицензия № 13С8-190514-084943-783-1256 от 15.05.2019
<b>6.3.2 Перечень информационных справочных систем и профессиональных баз данных</b>	
1.	Базы данных издательства Springer ( <a href="https://link.springer.com">https://link.springer.com</a> )
2.	Полнотекстовый архив ведущих западных научных журналов на российской платформе Национального электронно-информационного консорциума (НЭИКОН)( <a href="http://neicon.ru">http://neicon.ru</a> )
3.	Web of Science Core Collection – политематическая реферативно-библиографическая и наукометрическая (библиометрическая) база данных ( <a href="http://webofscience.com">http://webofscience.com</a> )
4.	Портал «Информационно-коммуникационные технологии в образовании» ( <a href="http://www.ict.edu.ru">http://www.ict.edu.ru</a> )
5.	Портал Федеральных государственных образовательных стандартов высшего образования ( <a href="http://fgosvo.ru">http://fgosvo.ru</a> )
6.	Официальный интернет-портал базы данных правовой информации ( <a href="http://pravo.gov.ru">http://pravo.gov.ru</a> )
7.	Компьютерная информационно-правовая система «Гарант»

### 7. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

Ауд.	Назначение	Оборудование и технические средства обучения	Вид
4-305	Компьютерный класс	аудиоколонки для проектора и интерактивной доски, аудиоколонки учебные, интерактивная доска, компьютеры, кондиционер, маркерная доска, проектор, столы компьютерные, столы учебные	
4-306	Компьютерный класс	аудиоколонки для проектора и интерактивной доски, интерактивная доска, компьютеры, кондиционер, маркерная доска, проектор, столы компьютерные, столы учебные	
4-307	Компьютерный класс	аудиоколонки, компьютеры, кондиционер, маркерная доска, столы компьютерные, столы учебные, телевизор	

### 8. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ (МОДУЛЯ)

Дисциплина «Практикум по базам 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 про-грамм;
- 5) Организация презентации индивидуальных заданий с целью формирования навыка владения профессиональной терминологией, грамотного выстраивания цепочки высказываний для доказательства обоснованности построенных моделей предметной области и правильно-сти 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.