

МИНПРОСВЕЩЕНИЯ РОССИИ

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

Базы данных

рабочая программа дисциплины (модуля)

Закреплена за кафедрой	институт передовых информационных технологий
ОПОП	Направление 02.03.02 Фундаментальная информатика и информационные технологии направленность (профиль) Инженерия программного обеспечения
Квалификация	Бакалавр
Год начала подготовки	2022
Форма обучения	очная
Общая трудоемкость	4 з.е.

Виды контроля по семестрам:

зачет 3

Семестр(Курс.Номер семестра на курсе)	3(2.1)		Итого	
	УП	РПД	УП	РПД
Лекции	34	34	34	34
Лабораторные	34	34	34	34
Итого ауд.	68	68	68	68
КСР	4	4	4	4
Контактная работа	72	72	72	72
Сам. работа	72	72	72	72
Часы на контроль	0	0	0	0
Практическая подготовка	0	0	0	0
Семинары	0	0	0	0
Консультации	0	0	0	0
Итого трудоемкость в часах	144	144	144	144

Программу составил(и):

д.п.н., профессор, Богатырева Юлия Игоревна

Рабочая программа дисциплины

Базы данных

разработана в соответствии с ФГОС:

Федеральный государственный образовательный стандарт высшего образования - бакалавриат по направлению подготовки 02.03.02 Фундаментальная информатика и информационные технологии (приказ Минобрнауки России от 23.08.2017 г. № 808)

составлена на основании учебного плана:

Направление 02.03.02 Фундаментальная информатика и информационные технологии
направленность (профиль) Инженерия программного обеспечения
утвержденного Учёным советом вуза от 28.02.2022 протокол № 3.

РПД утверждена Учёным советом университета
протокол от 28.2.2022 г. № 3

1. ЦЕЛИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

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

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ООП

Цикл (раздел) ООП:	Б1.О
2.1	Требования к предварительной подготовке обучающегося:
1.	Архитектура вычислительных систем
2.	Введение в профессию
3.	Программирование
4.	Введение в программирование
2.2	Дисциплины и практики, для которых освоение данной дисциплины (модуля) необходимо как предшествующее:
1.	Основы проектирования веб-интерфейсов
2.	Практикум по веб-программированию
3.	Разработка приложений для мобильных платформ
4.	Экономические информационные системы
5.	Применение нейронных сетей
6.	Разработка экономических информационных систем
7.	Экономико-правовые основы рынка программного обеспечения
8.	Аналитика больших данных
9.	Проектирование и разработка пользовательских интерфейсов

3. СООТНЕСЕНИЕ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ (МОДУЛЮ) С ИНДИКАТОРАМИ ДОСТИЖЕНИЯ КОМПЕТЕНЦИЙ

3.1 Компетенции обучающегося и индикаторы их достижения:

ОПК-3: Способен к разработке алгоритмических и программных решений в области системного и прикладного программирования, математических, информационных и имитационных моделей, созданию информационных ресурсов глобальных сетей, образовательного контента, прикладных баз данных, тестов и средств тестирования систем и средств на соответствие стандартам и исходным требованиям

ОПК-3.1	Знает методы теории алгоритмов, методы системного и прикладного программирования, основные положения и концепции в области математических, информационных и имитационных моделей
	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
ОПК-3.2	Умеет соотносить знания в области программирования, интерпретацию прочитанного, определять и создавать информационные ресурсы глобальных сетей, образовательного контента, средств тестирования систем
	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
ОПК-3.3	Имеет практический опыт применения разработки программного обеспечения
	владеет основными методами, способами и средствами получения, хранения, обработки информации в реляционных моделях данных владеет типовыми технологиями разработки баз данных

3.2 Результаты обучения по дисциплине:

В результате освоения дисциплины обучающийся должен:

	Знать:
3.1	- основные понятия в области теории баз данных
3.2	- основные технологии обработки баз данных
3.3	- этапы жизненного цикла баз данных
3.4	- виды и способы организации запросов к данным в реляционных моделях.
	Уметь:
У.1	- выбирать технологию для проектирования реляционной модели данных

У.2	- разрабатывать структуру модели данных
У.3	- реализовывать этапы жизненного цикла баз данных
У.4	- использовать технологии баз данных для обработки табличных данных
У.5	- представлять отчеты по результатам обработки данных средствами СУБД
Владеть:	
В.1	- владеет основными методами, способами и средствами получения, хранения, обработки информации в реляционных моделях данных
В.2	- владеет типовыми технологиями разработки баз данных

4. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

Код занятия	Наименование разделов и тем /вид занятия/	Семестр / Курс	Часов	Литература	Содержание
	Раздел 1. Теория баз данных				
1.1	Тема 1.1. Введение в предмет. Основные понятия. Модели данных /Лек/	3	2	Л1.2 Л1.3Л2.1 Л2.2 Л2.3	Сферы применения баз данных. История развития автоматизированных систем обработки дан-ных. 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.
1.2	Лабораторная работа 1. Введение в технологии баз данных и СУБД /Лаб/	3	2	Л1.2 Л1.3Л2.2 Л2.3	
1.3	Лабораторная работа 2. Реляционная модель данных /Лаб/	3	2	Л1.1 Л1.2 Л1.3Л2.2 Л2.3	
1.4	Тема 1.2. Представление данных в информационных системах /Лек/	3	4	Л1.2 Л1.3Л2.2 Л2.3 Л2.4	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
1.5	Лабораторная работа 3. Логическая модель данных /Лаб/	3	2	Л1.2 Л1.3Л2.2 Л2.3	
1.6	Лабораторная работа 4. Физическая модель данных /Лаб/	3	2	Л1.2 Л1.3Л2.2 Л2.3	
1.7	Представление данных в информационных системах /Ср/	3	8	Л1.2 Л1.3Л2.1 Л2.2 Л2.3	
1.8	Тема 1.3. Реляционная модель данных /Лек/	3	4	Л1.1 Л1.2 Л1.3Л2.1 Л2.2 Л2.3	Структуры данных реляционной модели. Использование отношений для представления данных. Ограничения модели. Функциональные зависимости, правила вывода функциональных зависи-мостей, полная функциональная зависимость. Ключи отношений.

	Раздел 2. Проектирование баз данных				
2.1	Тема 2.1. Этапы проектирования баз данных /Лек/	3	2	Л1.2 Л1.3Л2.2 Л2.3	Цели и задачи проектирования БД, жизненный цикл приложения БД. Этапы проектирования: определение требований к системе, анализ предметной области, работа с документами, пред-ставление результатов анализа, разработка реализаций, сравнение вариантов и выбор наилучшего, конструирование, эксплуатация и сопровождение БД
2.2	Тема 2.2. Метод нормальных форм /Лек/	3	2	Л1.2 Л1.3Л2.2 Л2.3	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.
2.3	Метод нормальных форм /Ср/	3	7	Л1.2 Л1.3Л2.2 Л2.3	
2.4	Тема 2.3. Метод ER-диаграмм /Лек/	3	2	Л1.2 Л1.3Л2.2 Л2.3	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.
2.5	Метод ER-диаграмм /Ср/	3	7	Л1.2 Л1.3Л2.2 Л2.3	
2.6	Тема 2.4. Инструментальные средства проектирования структур данных /Лек/	3	2	Л1.2 Л1.3Л2.2 Л2.3	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.
	Контроль самостоятельной работы студентов				
3.1	Контрольная работа /КСР/	3	2		Проверка теоретического материала
	Подготовка к зачету				
4.1	Подготовка к зачету /Ср/	3	18	Л1.2 Л1.3Л2.2 Л2.3	Вопросы к зачету

	Раздел 3. Язык запросов SQL				
5.1	Тема 3.1. Операторы определения данных /Лек/	3	4	Л1.2 Л1.3Л2.2 Л2.3	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.
5.2	Тема 3.2. Оператор выборки данных /Лек/	3	4	Л1.2 Л1.3Л2.1 Л2.2 Л2.3	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)
5.3	Лабораторная работа 2. Оператор выборки записей: выражения и функции в операторе SELECT /Лаб/	3	4	Л1.2 Л1.3Л2.2 Л2.3	
5.4	Тема 3.3. Оператор манипулирования данными /Лек/	3	6	Л1.2 Л1.3Л2.1 Л2.2 Л2.3	Полный и сокращенный синтаксис оператора добавления данных. Синтаксис операторов на обновление и удаление данных
5.5	Оператор манипулирования данными /Ср/	3	12	Л1.2 Л1.3Л2.2 Л2.3	
	Раздел 4. Программирование и администрирование баз данных				
6.1	Лабораторная работа 4. Транзакции. Целостность и восстановление баз данных /Лаб/	3	4	Л1.2 Л1.3Л2.2 Л2.3 Л2.4	
6.2	Лабораторная работа 5. Хранимые процедуры и триггеры /Лаб/	3	4	Л1.2 Л1.3Л2.2 Л2.3 Л2.4	
6.3	Тема 4.2. Защита баз данных; целостность и сохранность баз данных /Лек/	3	2	Л1.2 Л1.3Л2.1 Л2.2 Л2.3 Л2.4	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.
6.4	Лабораторная работа 6. Создание и удаление баз данных /Лаб/	3	2	Л1.2 Л1.3Л2.2 Л2.3 Л2.4	
6.5	Лабораторная работа 7. Управление учетными записями и правами доступа. Резервное копирование и восстановление баз данных /Лаб/	3	2	Л1.2 Л1.3Л2.2 Л2.3 Л2.4	
6.6	Защита баз данных; целостность и сохранность баз данных /Ср/	3	10	Л1.2 Л1.3Л2.2 Л2.3 Л2.4	
6.7	Лабораторная работа 9. Объектно-реляционный подход к моделированию систем. /Лаб/	3	2	Л1.1 Л1.2 Л1.3Л2.2 Л2.3	

6.8	Лабораторная работа 10. Средства идентификации объектов в модели и средства моделирования /Лаб/	3	2	Л1.1 Л1.2 Л1.3Л2.2 Л2.3	
6.9	Лабораторная работа 11. Оптимизация запросов в базе данных /Лаб/	3	2	Л1.1 Л1.2 Л1.3Л2.2 Л2.3	
6.10	Современные тенденции построения баз данных /Ср/	3	10	Л1.2 Л1.3Л2.2 Л2.3	
6.11	Лабораторная работа 12. Хешированные, индексированные файлы /Лаб/	3	2	Л1.2 Л1.3Л2.1 Л2.2 Л2.3	
6.12	Лабораторная работа 13. Логические структуры данных и структуры данных для приложений /Лаб/	3	2	Л1.2 Л1.3Л2.1 Л2.2 Л2.3	
	Контроль самостоятельной работы студентов				
7.1	Контрольная работа /КСП/	3	2		Вопросы по практической части

5. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ

5.1. Типовые задания для проведения текущего контроля

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

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 спи-сок 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 вос-хождений (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 вра-чей. В таблицы должны быть занесены имя, пол, дата рождения и домашний адрес каждого их па-циента. Всякий раз, когда врач осматривает больного, явившегося к нему на прием, или сам при-ходит 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 дан-ной 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, про-блемы, 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 ка-теров. 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 полу- чили 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, 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 опера-ции: 1) для указанного интервала дат вывести список аукционов с указанием наименования, даты и места проведения; 2) добавить на указанный пользователем аукцион на продажу предмет искус-ства с указанием начальной цены; 3) вывести список аукционов, с указанием суммарного дохода от продажи, отсортированных по доходу; 4) для указанного интервала дат, вывести список пред-метов, 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 фили-ала посчитать количество экземпляров указанной книги, находящихся в нем; 2) для указанной книги посчитать количество факультетов, на которых она используется в данном филиале, и вывести на- звания 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 ауди- торном 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 процедур 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 указанием 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 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 наклейка, 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, для каждого из них в базе заведена отдельная запись. Когда выходит новая пластинка, регистрируется название выпускающей 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, разработать 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 и 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 Vldel содержит информацию о владельцах лошадей. Эта таблица содержит следующие поля. Vldel_id – содержит идентификатор владельца и это поле объявлено как первичный ключ. Vldel_Name – содержит имя владельца. Vldel_Address – содержит адрес владельца. Vldel_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 Results_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 var-char(10) not null);

19.

20. Заполнение таблицы:

21. insert into Pol values (1,'man'); insert into Pol values (2,'woman'); Таблица IИпподромы: 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');

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, vldel_id integer, FOREIGN KEY (force_pol) REFERENCES pol, FOREIGN KEY (vldel_id) REFERENCES vldel);

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','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);

30. Таблица Владельцы:

31. create table vldel (vldel_id integer primary key, vldel_name varchar(20) not null, vldel_address varchar(40) not null, vldel_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'); 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,'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 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);

40. Заполнение таблицы:

41. 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)

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 CurForce1 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, содержащий 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, rezults_zabeg.rezult,rezults_zabeg.rez_time

52. FROM Fores,Gokey,rezults_zabeg

53. 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)

54. Процедура Lidars_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, rezults_zabeg.rez_time,zabeg.zabeg_data

56. FROM Forces,Gokey,rezults_zabeg,zabeg

57. WHERE (rezults_zabeg.rezult=1)AND (forces.force_id=rezults_zabeg.force_id)AND (gokey.gokey_id=rezults_zabeg.gokey_id)AND (zabeg.zabeg_id=rezults_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 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.

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 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 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 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;
69. Создание тела пакета:
70. 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 vla-
del_id=Arg; RETURN n; END; PROCEDURE Forces_Of_Vladel (Arg Vla-del.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 Forces.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
re-zults_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
forces.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; /
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 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
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 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; /

```

Вопросы к контрольным работам

1. Понятие базы данных. Предпосылки появления баз данных.
2. Понятие банк данных. Классификация банков данных и их отличия от базы данных.
3. Классификация баз данных по разным основаниям.
4. Сферы применения и перспективы развития баз данных.
5. Понятие системы управления базами данных. Классификация СУБД.
6. Понятие информационной системы и области ее применения.
7. Уровни представления данных в соответствии со стандартом ANSI.
8. Жизненный цикл базы данных. Выбор системы управления базами данных.
9. Функции СУБД. Основные категории и роли пользователей баз данных.
10. Понятие предметной области базы данных. Полная предметная область и организационная единица предметной области.
11. Операции над данными. Ограничения целостности. Управление данными.
12. Понятие и принципы построения иерархической модели данных. Примеры сбалансированного и несбалансированного графов
13. Понятие и принципы построения сетевой модели данных: ее недостатки и достоинства.
14. Основные принципы построения реляционной модели данных. Ее отличия от других моделей. Примеры,

достоинства и недостатки.

15. Объектно-ориентированная модель данных и принципы инкапсуляции, полиморфизма и наследование в модели.
16. Основные отличительные особенности постреляционной и многомерной моделей баз данных.
17. Полуструктурированная модель данных: примеры, достоинства и недостатки.
18. Понятия объекта базы данных, атрибут, сущность и связь. Связи между сущностями.
19. Понятие «транзакции». Свойства транзакции. Варианты завершения транзакции.
20. Этапы проектирования баз данных.
21. Модель «сущность-связь». Средства создания ER-модели. Пример проектирования БД методом «сущность-связь».
22. Нормализация отношений. Метод нормальных форм. 1, 2 и 3 нормальные формы.
23. Метод нормальных форм: НФБК, 4 и 5 нормальные формы. Основные преимущества нормализации
24. Схема данных. Первичные и составные ключи. Связи между отношениями.
25. Логическое проектирование. Преобразование инфологической модели в даталогическую
26. Физическая модель базы данных.
27. Этапы жизненного цикла баз данных.
28. Базовые понятия реляционной алгебры. Элементы реляционной модели.
29. Основные операции реляционных моделей. Понятие отношения. Свойства отношения.
30. Язык SQL: стандарт и реализация. Пример использования.
31. Язык SQL: основные категории. Типа данных и синтаксис.
32. Язык SQL: определение данных. Команды: CREATE TABLE; ALTER TABLE; RENAME TABLE; DROP TABLE; CREATE INDEX; DROP INDEX.
33. Язык SQL: операторы манипулирования данными.
34. Язык SQL: команды защиты и управления данными. Команды управления транзакциями.
35. Язык SQL: команда SELECT. Структура оператора. Простейшие и связанные запросы. Примеры использования.
36. Язык SQL: агрегатные функции. Примеры их применения.
37. Язык SQL: функция Like. Примеры ее применения.
38. СУБД с централизованной архитектурой. Примеры. Достоинства и недостатки.
39. СУБД с архитектурой файл-сервер. Примеры. Достоинства и недостатки.
40. СУБД с архитектурой клиент-сервер. Примеры. Достоинства и недостатки.
41. СУБД с трехуровневой архитектурой. Понятие Тонкий клиент - Сервер приложений - Сервер базы данных. Понятие SQL-сервера. Примеры СУБД с трехуровневой архитектурой.
42. Сравнение OLAP и OLTP технологий. Требования, предъявляемые к OLAP-системам.
43. Модели доступа и основные стратегии распределения данных, их достоинства и недостатки.
44. СУБД MySQL: обзор возможностей, структура БД MySQL, язык SQL.
45. Взаимодействие PHP и MySQL. Установка и завершение соединения. Обработка запросов.
46. Основные функции PHP для работы с MySQL.
47. Понятие «администратор БД». Функционал администратора БД.
48. Администрирование в MySQL.
49. Защита данных и администрирование приложений. Классификация угроз безопасности работы БД. Способы защиты информации в БД. Механизмы обеспечения безопасности баз данных.
50. Категории пользователей БД. Система назначения полномочий. Роль и привилегии у различных категорий

5.2. Типовые задания для проведения промежуточной аттестации

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.

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

5.3. Перечень видов оценочных средств

1. Зачет
2. Вопросы к контрольным работам
3. Индивидуальное задание по лабораторным работам

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 "БД_Ваныкина.pdf"

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

Максимальная сумма баллов – 100.

Промежуточная аттестация – 70 баллов, зачет – 30 баллов.

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, пропустивший занятия в ходе се-местра, 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.

6. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

6.1. Рекомендуемая литература

6.1.1. Основная литература

	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	http://biblioclub.ru/index.php?page=book&id=233290
Л1.2	Гущин А. Н.	Базы данных: учебно-методическое пособие	, 2015	http://biblioclub.ru/index.php?page=book&id=278093
Л1.3	Карпова Т. С.	Базы данных: модели, разработка, реализация : учебное пособие	, 2016	http://biblioclub.ru/index.php?page=book&id=429003

6.1.2. Дополнительная литература

	Авторы, составители	Заглавие	Издательство, год (кол-во экземпляров для печатных изданий)	Ссылка на электронное издание
Л2.1	Чубукова И. А.	Data Mining	, 2008	http://biblioclub.ru/index.php?page=book&id=233055
Л2.2	Медведкова И. Е., Бугаев Ю. В., Чикунов С. В.	Базы данных	, 2014	http://biblioclub.ru/index.php?page=book&id=336032
Л2.3	Сирант О. В., Коваленко Т. А.	Работа с базами данных	, 2016	http://biblioclub.ru/index.php?page=book&id=428978
Л2.4	Кияев В., Граничин О.	Безопасность информационных систем: курс	Москва: Национальный Открытый Университет «ИНТУИТ», 2016	http://biblioclub.ru/index.php?page=book&id=429032

6.2. Перечень ресурсов информационно-телекоммуникационной сети "Интернет"

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.			

6.3. Информационные технологии

6.3.1 Перечень лицензионного и свободно распространяемого программного обеспечения

1.	Операционная система Microsoft Windows Professional 7 Russian. Лицензия №48497058 от 13.05.2011 г., договор № Пр/16/6 от 05 апреля 2016 г.			
2.	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.			
3.	Программное обеспечение Microsoft Office Enterprise 2007 Russian. Лицензия №46138962 от 16.11.2009			
4.	Программное обеспечение Microsoft Office 2013 Professional. Контракт № 405535 от 2 ноября 2015 года, контракт № ПР/ФЕН/15/18 от 23.10.2015 г.			
5.	Браузеры Google Chrome, Mozilla, Opera. Свободно распространяемое ПО			
6.	Операционная система Microsoft Windows XP Professional Russian. Лицензия № 16698685 от 08.08.2003 г.			
7.	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.			
8.	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.			
9.	Комплексная система антивирусной защиты Kaspersky Endpoint Security для бизнеса – стандартный Russian Edition. 500-999 Node 2 year Educational Renewal License. Лицензия № 13C8-190514-084943-783-1256 от 15.05.2019			

6.3.2 Перечень информационных справочных систем и профессиональных баз данных

1.	Компьютерная информационно-правовая система «Гарант»			
2.	Официальный интернет-портал базы данных правовой информации (http://pravo.gov.ru)			
3.	Портал Федеральных государственных образовательных стандартов высшего образования (http://fgosvo.ru)			
4.	Портал «Информационно-коммуникационные технологии в образовании» (http://www.ict.edu.ru)			
5.	Web of Science Core Collection – политематическая реферативно-библиографическая и наукометрическая (библиометрическая) база данных (http://webofscience.com)			
6.	Полнотекстовый архив ведущих западных научных журналов на российской платформе Национального электронно-информационного консорциума (НЭИКОН)(http://neicon.ru)			
7.	Базы данных издательства Springer (https://link.springer.com)			

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	Лаб

Ауд.	Назначение	Оборудование и технические средства обучения	Вид
4-306	Компьютерный класс	аудиоколонки для проектора и интерактивной доски, интерактивная доска, компьютеры, кондиционер, маркерная доска, проектор, столы компьютерные, столы учебные	Лек
4-307	Компьютерный класс	аудиоколонки, компьютеры, кондиционер, маркерная доска, столы компьютерные, столы учебные, телевизор	Ср
2-15	Компьютерный класс	компьютеры, рулонный экран, стол преподавателя, столы компьютерные, переносной проектор	КСР
2-16	Компьютерный класс	интерактивная доска, компьютеры, маркерная доска, принтер, сканер, стол преподавателя, столы учебные	Зачёт

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

Дисциплина «Базы 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 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.