**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ**

**РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФГБОУ ВО «СГУ имени Н.Г. Чернышевского»**

Механико-математический факультет

|  |  |
| --- | --- |
| СОГЛАСОВАНОзаведующий кафедройматематического и компьютерного моделированияЮ.А. Блинков"\_\_" \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_\_ г. | УТВЕРЖДАЮпредседатель НМС факультетаС.В. Тышкевич"\_\_" \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_\_ г. |

**Фонд оценочных средств**

Текущего контроля и промежуточной аттестации по дисциплине

**Базы данных**

Направление подготовки бакалавриата

**01.03.02 Прикладная математика и информатика**

Профиль подготовки бакалавриата

**Математическое моделирование,**

**Математическое и информационное обеспечение экономической деятельности,**

**Математическая физика и современные компьютерные технологии**

Квалификация (степень) выпускника

*Бакалавр*

Форма обучения

*очная*

Саратов,

**2016 год**

***Карта компетенций***

|  |  |
| --- | --- |
| **Контролируемые компетенции****(шифр компетенции)** | **Планируемые результаты обучения****(знает, умеет, владеет, имеет навык)** |
|
| ОК-6 Способность работать в команде, толерантно воспринимая социальные, этнические, конфессиональные и культурные различия | **Знать:**принципы функционирования профессионального коллектива, понимать роль корпоративных норм и стандартов**З (ОК-6)-I**о социальных, этнических, конфессиональных и культурных особенностях представителей тех или иных социальных общностей**З (ОК-6) –II** |
| **Уметь:**работать в коллективе, эффективно выполнять задачи профессиональной деятельности**У (ОК-6)-I**работая в коллективе, учитывать социальные, этнические, конфессиональные, культурные особенности представителей различных социальных общностей в процессе профессионального взаимодействия в коллективе, толерантно воспринимать эти различия**У (ОК-6) –II** |
| **Владеть:**приемами взаимодействия с сотрудниками, выполняющими различные профессиональные задачи и обязанности**В (ОК-6)-I**в процессе работы в коллективе этическими нормами, касающимися социальных, этнических, конфессиональных и культурных различий; способами и приемами предотвращения возможных конфликтных ситуаций в процессе профессиональной деятельности**В (ОК-6) –II** |
| ОПК-3способность 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. | **Знать:**знать основные понятия, концепции, результаты, задачи и методы программирования, построения баз данных и графических моделей, знать результаты, задачи и методы информатики.**Код З (ОПК-3) – I**знать основные понятия теории объектно-ориентированное программирования обыкновенных и методы численного анализа, а также теорию построения и анализа баз данных, определения и свойства математических объектов в этих областях, формулировки ключевых утверждений, методы их доказательства, возможные сферы их приложений.**Код З (ОПК-3) – II**знать основные понятия, концепции, результаты, задачи и методы численного анализа вероятностей, построения баз данных и парадигм программирования.**Код З (ОПК-3) – III** |
| **Уметь:**уметь применять основные методы анализа к исследованию и созданию баз данных, не умеет имплементировать стандартные численные алгоритмы, не умеет создавать простейшие графические модели.**Код У (ОПК-3) – I**уметь решать задачи вычислительного характера в различных областях численного анализа, используя объектно-ориентированное программирование.**Код У (ОПК-3) – II**уметь применять математические методы и модели к численному анализу для их описания и понимания, а также применять численные методы для решения задач из различных областей математики.**Код У (ОПК-3) – III** |
| **Владеть:**владеть навыками построения графических моделей, разработки базовых алгоритмов в различных языках программирования, создания и анализа баз данных.**Код В (ОПК-3) – I**навыками объектно-ориентированного программирования в различных языках программирования.**Код В (ОПК-3) – II**владеть навыками программирования сложных алгоритмов с использованием объектно-ориентированной парадигмы программирования, а также решения простых алгоритмов с использованием функциональной парадигмы программирования.**Код В (ОПК-3) – III** |
| ОПК-4 Способность решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности | **Знать:**методы сбора, обработки и хранения информации, а также основные методы формирования научного знания.**Код З (ОПК-4) – I**классификацию языков программирования, основные методы разработки программного обеспечения, стандарты оформления программной документации и причины нарушения компьютерной безопасности.**Код З (ОПК-4) – II** |
| **Уметь:**использовать научные и методические ресурсы сети интернет для разработки программного обеспечения и программной документации с учетом требований информационной безопасности.**Код У (ОПК-4) – I**составлять научные обзоры, рефераты и библиографии по тематике научных исследований.**Код У (ОПК-4) – II** |
| **Владеть:**базовыми навыками по защите информации на рабочем месте, в корпоративных сетях при входе в глобальные сети.**Код В (ОПК-4) – I**навыками системного и объектно-ориентированного программирования для решения стандартных прикладных задач в профессиональной деятельности.**Код В (ОПК-4) – II** |
| ПК-1 способность собирать, обрабатывать и интерпретировать данные современных научных исследований, необходимые для формирования выводов по соответствующим научным исследованиям. | **Знать:**определения и свойства интегралов Римана и Лебега, признаки сходимости функциональных рядов, свойства степенных рядов, преобразование Фурье и его свойства, основные типы обыкновенных уравнений, метод Эйлера, метод вариации произвольных постоянных, основные понятия теории устойчивости, формулы Крамера, жорданову форму матрицы, уравнения кривых второго порядка и их свойства, поверхности второго порядка, основные математические модели дискретного характера и методы их использования для решения типовых задач;основные понятия, методы и средства обработки информации, теоретические основы поиска информации, классификацию информационных ресурсов по способу представления информации.**З (ПК-1) — I**определения статических и динамических структур;понятия глобальных объектов и констант;технологии программирования;основы архитектуры операционных систем;способы оптимизации передачи данных;способы обеспечения безопасности в сетях;основы архитектуры параллельных вычислительных;классификацию уровней математической физики второго порядка;метод Даламбера решения задачи Коши для волнового уравнения;свойства гармонических функций;основные принципы функционального анализа;теорию Фредгольма для операторных уравнений;классификацию особых точек аналитических функций;теорию вычетов;формулу Коши для аналитических функций;принципы построения и проектирования баз данных;язык баз данных SQL.**З (ПК-1) – II**понятия регулярного и сингулярного возмущений;понятие асимптотического ряда типа Пуанкаре;понятия высшего приближения, внутреннего и внешнего разложений;основные понятия теории классических симметрий для дифференциальных уравнений;законы сохранения;технологии программирования;основные понятия системного анализа;особенности экономических систем;основные методы разработки и исследования математических, информационных и имитационных моделей;теорему Куна-Таккера для задач выпуклого программирования;теорию двойственности в линейном программировании; принцип максимума Понтрягина.**З (ПК-1) - III** |
| **Уметь:**осуществлять поиск научной информации в интернете, использовать основные понятия и методы математического анализа, дифференциальных уравнений, дискретной математики, алгебры, геометрии и информатики при обработке и интерпретации собранных данных.**У (ПК-1) — I**использовать парадигму объектной технологии;уметь использовать принцип модульности в математическом моделировании;выбирать адекватные методы моделирования систем;разрабатывать и отлаживать эффективные алгоритмы и программы с использованием современных технологий программирования;обрабатывать исключения в случае нарушения контракта;решать краевые задачи для различных типов уравнений математической физики методом разделения переменных;решать задачу Коши для уравнений параболического типа;находить нормы функционалов и оценивать нормы операторов;строить конформные отображения простых областей с помощью элементарных аналитических функций;осуществлять поиск заданной информации в системах Yandex и Google с использованием профессионального языка запросов.**У (ПК-1) – II**вычислять симметрии дифференциальных уравнений средствами компьютерной алгебры;получать асимптотические разложения для конкретных функций;применять методы Линдштедта-Пуанкаре и Лайтхилла для получения решений дифференциальных уравнений строить модели данных DFD;применять основные принципы и законы моделирования систем;уметь строить UML модели экономических процессов;находить двойственные задачи и задачи линейного программирования;решать экстремальные задачи методом золотого сечения и методом наискорейшего спуска.**У (ПК-1) – III** |
| **Владеть:**навыками сбора и работы с математическими источниками информации, методами математического анализа и навыками их практического применения: навыками дифференцирования функций, методами решения линейных дифференциальных уравнений, методами решения систем линейных алгебраических уравнений, навыками работы с компьютером как средством управления информацией, навыками работы в OC WINDOWS.**В (ПК-1) – I**навыками приведения уравнений в частных производных второго порядка к каноническому виду, методом разделения переменных;навыками применения основных понятий и методов функционального анализа при исследовании конкретных задач;навыками разложения аналитических функций в ряд Лорана, вычисления контурных интегралов с помощью теории вычетов;навыками работы в среде систем управления базами данных, методами сбора и обработки информации, навыками работы с поисковыми системами;навыками моделирования прикладных задач методами дискретной математики;навыками работы с инструментами системного анализа;комбинаторным, теоретико-множественным и вероятностным подходами к постановке и решению задач;навыками программирования в современных средах.**В (ПК-1) – II**навыками построения асимптотических рядов для конкретных функций;усовершенствованной процедурой сращивания;методами генерации разностных схем с алгебраическими свойствами дифференциальных уравнений;методами моделирования информационных процессов;основными методами разработки алгоритмических и программных решений в области системного и прикладного программирования, математических, информационных и имитационных моделей;методом множителей Лагранжа при решении задач на условный экстремум;симплекс-методом при решении канонических задач линейного программирования.**В (ПК-1) – III** |
| ПК-4 Способность работать в составе научно-исследовательского и производственного коллектива и решать задачи профессиональной деятельности | **Знать:**принципы построения и проектирования баз данных, принципы функционирования систем управления базами данных, знать основы языка баз данных SQL; \_\_З (ПК-4) –I |
| **Уметь:**применять полученные знания в практике проектирования и эксплуатации баз данных, проводить системный анализ предметной области; разрабатывать интерфейс баз данных с использованием современных программных инструментальных средств; работать в составе научно-исследовательского и производственного коллективов; \_\_У (ПК-4) –I |
| **Владеть:**Навыками использования CASE-средств при моделировании прикладных задач и проектировании схем баз данных; навыками формулирования запросов к базам данных на языке SQL; навыками работы в среде систем управления базами данных; \_\_В (ПК-4) –I |
| ПК-5 способность осуществлять целенаправленный поиск информации о новейших научных и технологических достижениях в информационно - телекоммуникационной сети «Интернет» и в других источниках. | **Знать:**Основные математические модели, описывающие физические, химические, биологические, социальные, экономические процессы и явления, и приводящие к дифференциальным уравнениям; способы описания движения и основные характеристики состояния сплошной среды, общий принцип построения математических моделей и простейшие математические модели сс; \_\_З (ПК-5) –I |
| **Уметь:**Выбирать математическую модель для изучаемых процессов; моделировать и выбирать метод решения поставленной задачи с использованием современной вычислительной техники; \_\_У (ПК-5) –I |
| **Владеть*:***Методами математического моделирования при изучении объектов различной природы; способностью самостоятельно изучать и анализировать научную и методическую литературу, связанную с проблемами МСС; \_\_В (ПК-5) –I |
| ПК-7 Способность к разработке и применению алгоритмических и программных решений в области системного и прикладного программного обеспечения | **Знать:**основные понятия, идеи, методы, связанные с дисциплинами фундаментальной математики, информатики, математического моделирования ;  краткую историю эволюции вычислительных систем; технологии программирования, основы архитектуры операционных систем; задачи и методы исследования и обеспечения качества и надежности программных компонентов. \_З (ПК-7) –Iсовременное состояние, тенденции и наиболее важные проблемыразвития системного и прикладного программного обеспечения;методологию построения математических алгоритмов, методы компьютерной геометрии и геометрического моделирования, основные языки программирования и методы трансляции. \_З (ПК-7) –IIнавыками разработки алгоритмических и программных решений в области системного и прикладного программного обеспечения; практического анализа логики различного рода рассуждений, аргументации, ведения дискуссий и полемики; доказывать оптимальность выбранного алгоритма, метода, объясняя его задачи и функции; навыками работы с инструментальными средствами проектирования баз данных и знаний, управления проектами информационных систем ; методикой объектно-ориентированного программирования; профессиональной терминологией при презентации построенных моделей. \_\_В (ПК-7) –III |
| **Уметь:**систематизировать методы фундаментальной математики для построения математических моделей в элементарных прикладных задачах, описывать основные этапы построения алгоритмов; разрабатывать и отлаживать эффективные алгоритмы и программы с использованием современных технологий программирования ; формулировать требования к создаваемым программным комплексам. \_У (ПК-7) –Iсамостоятельно осуществлять поиск специальной литературы и выбирать эффективные методы решения согласно поставленным задачам;в соответствии с выбранными методами решения строить математическую модель с алгоритмом ее реализации, оценивать погрешность получаемого решения;строить математические алгоритмы и реализовывать их с помощью языков программирования, применять методы математического моделирования к решению конкретных задач.У (ПК-7) –IIразрабатывать и применять алгоритмические и программные решения в области системного и прикладного программного обеспечения;публично представлять, объяснять, защищать построенную математическую модель и выбранный алгоритм; проектировать базы данных; создавать приложения, используя модульный и объектно-ориентированный подходобъяснять учебный и научный материал, вести корректную дискуссию в процессе представления математической модели и алгоритмов. \_\_У (ПК-7) –III |
| **Владеть:**методологией математического моделирования, навыками сбора и работы с математическими источниками информации, теоретическими основами построения алгоритмов; навыками работы с инструментами системного анализа; комбинаторным, теоретико-множественным и вероятностным подходами к постановке и решению задач; навыками программирования в современных средах. \_В (ПК-7) –Iнавыками систематизации и выбора необходимой информации согласно поставленной задачи, основными методами математического и алгоритмического моделирования ; навыками построения и реализации основных математических алгоритмов, основными языками программирования; навыками разработки приложений с использованием выбранной операционной системы и среды разработки; методами управления аппаратно-программными системами и комплексами, эксплуатации сложно организованных программных систем. \_В (ПК-7) –IIнавыками разработки алгоритмических и программных решений в области системного и прикладного программного обеспечения; практического анализа логики различного рода рассуждений, аргументации, ведения дискуссий и полемики; доказывать оптимальность выбранного алгоритма, метода, объясняя его задачи и функции; навыками работы с инструментальными средствами проектирования баз данных и знаний, управления проектами информационных систем ; методикой объектно-ориентированного программирования; профессиональной терминологией при презентации построенных моделей. \_\_В (ПК-7) –III |

***Показатели оценивания планируемых результатов обучения***

|  |  |
| --- | --- |
| **Семестр** | **Шкала оценивания** |
| **2** | **3** | **4** | **5** |
| **6 семестр** | Студент не владеет методами построения, сопровождения и модификации баз данных в соответствии с нуждами конечного пользователя, основанными на методах реляционной алгебры; языками программирования, ориентированными на обработку данных;не умеет применять методы реляционной алгебры и языки программирования, ориентированными на обработку данных для построения, сопровождения и модификации баз данных в соответствии с нуждами конечного пользователя;не знает основные принципы построения баз данных, операции реляционной алгебры, связанные с ними правила и теоремы, их реализацию и ее особенности в языках программирования ориентированных на обработку данных.Не владеет методами поиска и отбора литературы, наиболее соответствующей заданной тематике;не умеет составлять библиографический список по заданной тематике; не знаетгосударственные стандарты в области оформления библиографических ссылок и списков | Студент плохо владеет методами построения, сопровождения и модификации баз данных в соответствии с нуждами конечного пользователя, основанными на методах реляционной алгебры; языками программирования, ориентированными на обработку данных;с трудом может применять методы реляционной алгебры и языки программирования, ориентированными на обработку данных для построения, сопровождения и модификации баз данных в соответствии с нуждами конечного пользователя;не достаточно хорошо понимает основные принципы построения баз данных, операции реляционной алгебры, связанные с ними правила и теоремы, их реализацию и ее особенности в языках программирования ориентированных на обработку данных.Плохо владеет методами поиска и отбора литературы, наиболее соответствующей заданной тематике; с трудом может составлять библиографический список по заданной тематике; не достаточно хорошо знает государственные стандарты в области оформления библиографических ссылок и списков | Студент достаточно хорошо владеет методами построения, сопровождения и модификации баз данных в соответствии с нуждами конечного пользователя, основанными на методах реляционной алгебры; языками программирования, ориентированными на обработку данных;хорошо применяет методы реляционной алгебры и языки программирования, ориентированными на обработку данных для построения, сопровождения и модификации баз данных в соответствии с нуждами конечного пользователя;показано хорошее понимание основных принципов построения баз данных, операций реляционной алгебры, связанных с ними правил и теорем, их реализацию и ее особенностей в языках программирования ориентированных на обработку данных.Достаточно хорошо владеет методами поиска и отбора литературы, наиболее соответствующей заданной тематике;хорошо составляет библиографический список по заданной тематике;показано хорошее знание государственных стандартов в области оформления библиографических ссылок и списков. | Студент отлично владеет методами построения, сопровождения и модификации баз данных в соответствии с нуждами конечного пользователя, основанными на методах реляционной алгебры; языками программирования, ориентированными на обработку данных;уверенно применяет методы реляционной алгебры и языки программирования, ориентированными на обработку данных для построения, сопровождения и модификации баз данных в соответствии с нуждами конечного пользователя;уверенно ориентируется основных принципах построения баз данных, операциях реляционной алгебры, связанных с ними правилах и теоремах, их реализацию и ее особенности в языках программирования ориентированных на обработку данных.Отлично владеет методами поиска и отбора литературы, наиболее соответствующей заданной тематике;уверенно справляется с составлением библиографических списков по заданной тематике;уверенно ориентируется в государственных стандартах в области оформления библиографических ссылок и списков |

**Оценочные средства**

**Задания для текущего контроля**

Текущий контроль качества усвоения учебного материала по данной дисциплине методически целесообразно проводить в виде тестирования.

**Тестирование**

Методические указания. Тесты для текущего контроля выполняются в письменном виде с ограничением времени: по сорок секунд на задание.

Критерии оценивания. Уровень выполнения текущих тестовых заданий оценивается в баллах, которые затем переводятся в оценку. Баллы выставляются следующим образом:

* за каждый верно выбранный вариант ответа — 1 балл;
* максимально возможное количество баллов — 228.

Оценка соответствует следующей шкале:

|  |  |  |
| --- | --- | --- |
| Оценка | Кол-во баллов | Процент верных ответов |
| Отлично | 182 и более | Свыше 80 % |
| Хорошо | 149 - 181 | 66 – 80 % |
| Удовлетворительно | 114-148 | 50 – 65 % |
| Неудовлетворительно | менее 114 | менее 50 % |

***Тестовые задания:***

1. Модель жизненного цикла БД, наиболее неприемлемая:

1 каскадная модель

 0 итерационная модель

 0 спиральная модель

1. Модель жизненного цикла БД, наиболее приемлемая:

 0 каскадная модель

 0 итерационная модель

 1 спиральная модель

1. Последовательность моделей БД:

 0 реляционные, объектно-ориентированные, иерархические

 0 объектно-ориентированные, иерархические, реляционные

 1 иерархические, реляционные, объектно-ориентированные

 0 объектно-ориентированные, реляционные, иерархические

1. Иерархическое поколение БД наступило с

 1 1968 года

 0 1930 года

 0 1945 года

 0 1995 года

1. Реляционное поколение БД наступило с

 0 1999 года

 1 1970 года

 0 1995 года

 0 1968 года

1. Модель для описания иерархических структур, наиболее адекватная из:

 0 сетевая

 0 реляционная

 0 постреляционная

 0 объектно-ориентированная

 1 XML как модель данных

 0 многомерная модель данных (OLAP)

1. К управлению данными во внешней памяти, как функции СУБД, относится фраза:

 0 средства для работы с базами данных

 0 размер баз данных превышает размер оперативной памяти

 1 поддержка структур памяти для хранения данных и служебных целей

 0 поддержка логической целостности хранимых данных

 0 надежность хранения данных во внешней памяти

1. К управлению буферами оперативной памяти, как функции СУБД, относится фраза:

 0 надежность хранения данных во внешней памяти

 0 поддержка структур памяти для хранения данных и служебных целей

 0 поддержка логической целостности хранимых данных

 0 средства для работы с базами данных

 1 размер баз данных превышает размер оперативной памяти

1. К управлению транзакциями, как функции СУБД, относится фраза:

 0 средства для работы с базами данных

 0 размер баз данных превышает размер оперативной памяти

 1 поддержка логической целостности хранимых данных

 0 поддержка структур памяти для хранения данных и служебных целей

 0 надежность хранения данных во внешней памяти

1. К журнализации и восстановлению БД после сбоев, как функции СУБД, относится фраза:

 0 размер баз данных превышает размер оперативной памяти

 1 надежность хранения данных во внешней памяти

 0 поддержка структур памяти для хранения данных и служебных целей

 0 средства для работы с базами данных

 0 поддержка логической целостности хранимых данных

1. К поддержке языков баз данных, как функции СУБД, относится фраза:

 0 размер баз данных превышает размер оперативной памяти

 0 поддержка логической целостности хранимых данных

 1 средства для работы с базами данных

 0 поддержка структур памяти для хранения данных и служебных целей

 0 надежность хранения данных во внешней памяти

1. Ядро СУБД отвечает за функции:

 1 управление данными во внешней памяти

 1 управление буферами оперативной памяти

 1 управление транзакциями

 1 журнализация и восстановление БД после сбоев

 0 поддержка языков баз данных

1. Фраза, относящаяся к модели взаимодействия с БД с централизованной архитектурой:

 0 тонкие клиенты взаимодействуют с серверной СУБД через сервер приложений

 0 компьютеры вне сети имеют СУБД и реплики БД

 0 на файловом сервере хранятся файлы базы данных, на клиентских компьютерах –СУБД и клиентские приложения

 0 на сервере сети размещаются СУБД и файлы базы данных, на клиентских компьютерах – клиентские приложения

 0 поддерживается модулем СУБД, называемым репликатором

 1 СУБД и приложение располагаются на одном компьютере

 0 фрагменты базы данных могут размещаться на различных компьютерах

1. Фраза, относящаяся к модели взаимодействия с БД с автономными персональными компьютерами:

 0 поддерживается модулем СУБД, называемым репликатором

 0 СУБД и приложение располагаются на одном компьютере

 0 компьютеры вне сети имеют СУБД и реплики БД

 0 фрагменты базы данных могут размещаться на различных компьютерах

0 тонкие клиенты взаимодействуют с серверной СУБД через сервер приложений

 1 на файловом сервере хранятся файлы базы данных, на клиентских компьютерах – СУБД и клиентские приложения

 0 на сервере сети размещаются СУБД и файлы базы данных, на клиентских компьютерах – клиентские приложения

1. Фраза, относящаяся к архитектуре «файл-сервер»:

 0 СУБД и приложение располагаются на одном компьютере

 1 на файловом сервере хранятся файлы базы данных, на клиентских компьютерах – СУБД и клиентские приложения

 0 на сервере сети размещаются СУБД и файлы базы данных, на клиентских компьютерах – клиентские приложения

 0 компьютеры вне сети имеют СУБД и реплики БД

 0 фрагменты базы данных могут размещаться на различных компьютерах

 0 тонкие клиенты взаимодействуют с серверной СУБД через сервер приложений

 0 поддерживается модулем СУБД, называемым репликатором

1. Фраза, относящаяся к архитектуре «клиент-сервер»:

 0 компьютеры вне сети имеют СУБД и реплики БД

 0 СУБД и приложение располагаются на одном компьютере

 0 фрагменты базы данных могут размещаться на различных компьютерах

 0 на файловом сервере хранятся файлы базы данных, на клиентских компьютерах –СУБД и клиентские приложения

 0 поддерживается модулем СУБД, называемым репликатором

 1 на сервере сети размещаются СУБД и файлы базы данных, на клиентских компьютерах – клиентские приложения

 0 тонкие клиенты взаимодействуют с серверной СУБД через сервер приложений

1. Фраза, относящаяся к трехзвенной архитектуре «клиент-сервер»:

 0 фрагменты базы данных могут размещаться на различных компьютерах

 1 тонкие клиенты взаимодействуют с серверной СУБД через сервер приложений

 0 СУБД и приложение располагаются на одном компьютере

 0 на файловом сервере хранятся файлы базы данных, на клиентских компьютерах –СУБД и клиентские приложения

 0 поддерживается модулем СУБД, называемым репликатором

 0 компьютеры вне сети имеют СУБД и реплики БД

 0 на сервере сети размещаются СУБД и файлы базы данных, на клиентских компьютерах – клиентские приложения

1. Фраза, относящаяся к распределенным базам данных:

 0 компьютеры вне сети имеют СУБД и реплики БД

 0 на сервере сети размещаются СУБД и файлы базы данных, на клиентских компьютерах – клиентские приложения

 0 на файловом сервере хранятся файлы базы данных, на клиентских компьютерах – СУБД и клиентские приложения

 0 тонкие клиенты взаимодействуют с серверной СУБД через сервер приложений

 0 поддерживается модулем СУБД, называемым репликатором

 1 фрагменты базы данных могут размещаться на различных компьютерах

 0 СУБД и приложение располагаются на одном компьютере

1. Фраза, относящаяся к технологии тиражирования данных:

 0 фрагменты базы данных могут размещаться на различных компьютерах

 0 тонкие клиенты взаимодействуют с серверной СУБД через сервер приложений

 1 поддерживается модулем СУБД, называемым репликатором

 0 СУБД и приложение располагаются на одном компьютере

 0 компьютеры вне сети имеют СУБД и реплики БД

 0 на файловом сервере хранятся файлы базы данных, на клиентских компьютерах – СУБД и клиентские приложения

 0 на сервере сети размещаются СУБД и файлы базы данных, на клиентских компьютерах – клиентские приложения

1. Синоним понятия «пустое значение»:

 0 неизвестное значение

 1 одно из значений типа данных

 0 неприменимое значение

 0 неопределенное значение

 0 нулевое значение

1. Пустыми являются значения:

 1 0

 0 1

 1 false

 0 true

 1 строка переменной длины, не содержащая символов

 0 строка переменной длины, содержащая пробелы

 1 строка постоянной длины, содержащая пробелы

 0 null

1. Пустые значения

 1 могут иметь некоторые типы данных

 1 могут иметь все типы данных

 0 должны иметь некоторые типы данных

 0 должны иметь все типы данных

1. Null-значение может быть присвоено переменным

 1 любых типов

 0 некоторых типов

 0 только числовых типов

1. Интерпретация null-значения:

 1 значение пока неизвестно

 1 значение неизвестно

 1 значение неприменимо

 1 значение пока неприменимо

 0 значение пусто

 1 может изменяться с течением времени

 1 может быть неясной

 0 не зависит от семантики данных

1. Интерпретация null-значений в контексте арифметических операций:

 0 значение пока неизвестно

 0 значение неизвестно

 1 значение неприменимо

 1 значение пока неприменимо

 0 значение пусто

 0 может быть неясной

1. Интерпретация null-значений в контексте логических операций:

 1 значение пока неизвестно

 1 значение неизвестно

 0 значение неприменимо

 0 значение пока неприменимо

 0 значение пусто

 0 может быть неясной

1. Интерпретация null-значений в контексте строковых операций:

 0 значение пока неизвестно

 0 значение неизвестно

 1 значение неприменимо

 1 значение пока неприменимо

 0 значение пусто

 0 может быть неясной

1. Интерпретация null-значений в контексте операций сравнения:

 0 значение пока неизвестно

 0 значение неизвестно

 1 значение неприменимо

 1 значение пока неприменимо

 0 значение пусто

 0 может быть неясной

1. Переменная x имеет null-значение. Значение выражения «1 + x»:

 0 0

 0 1

 1 null

 0 false

 0 true

1. Переменная x имеет null-значение. Значение выражения «x - x»:

 0 0

 0 1

 1 null

 0 false

 0 true

1. Переменная x имеет null-значение. Значение выражения «0 \* x»:

 0 0

 0 1

 1 null

 0 false

 0 true

1. Переменная x имеет null-значение. Значение выражения «1 / x»:

 0 0

 0 1

 1 null

1. 0 false

 0 true

 0 ошибка

1. Переменная x имеет null-значение. Значение выражения «false and x»:

 0 0

 0 null

 1 false

 0 true

1. Переменная x имеет null-значение. Значение выражения «true and x»:

 0 0

 1 null

 0 false

 0 true

1. Переменная x имеет null-значение. Значение выражения «false or x»:

 0 0

 1 null

 0 false

 0 true

1. Переменная x имеет null-значение. Значение выражения «true or x»:

 0 0

 0 null

 0 false

 1 true

1. Переменная x имеет null-значение. Значение выражения «x or not x»:

 0 0

 1 null

 0 false

 0 true

1. Переменная x имеет null-значение. Значение выражения «'A' + x + 'B'»:

 0 'AB'

 0 'A B'

 0 'AnullB'

 1 null

 0 ''

1. Переменная x имеет null-значение. Значение выражения «x < null»:

 0 0

 1 null

 0 false

 0 true

1. Переменная x имеет null-значение. Значение выражения «x < x»:

 0 0

 1 null

 0 false

 0 true

1. Значение выражения «IsNull(0)»:

 0 0

 0 1

 0 null

 1 false

 0 true

1. Значение выражения «IsNull(1)»:

 0 0

 0 1

 0 null

 1 false

 0 true

1. Значение выражения «IsNull(1 / null)»:

 0 0

 0 1

 0 null

 0 false

 1 true

 0 ошибка

1. Значение выражения «IsNull(null / 0)»:

 0 0

 0 1

 0 null

 0 false

 1 true

 0 ошибка

1. Значение выражения «IsNull(false)»:

 0 0

 0 1

 0 null

 1 false

 0 true

1. Значение выражения «IsNull(true)»:

 0 0

 0 1

 0 null

 1 false

 0 true

1. Значение выражения «IsNull(not null)»:

 0 0

 0 1

 0 null

 0 false

 1 true

1. Значение выражения «IsNull('ABC' + null)»:

 0 0

 0 1

 0 null

 0 false

 1 true

 0 ошибка

 0 'ABC'

 0 'ABCnull'

 0 ''

1. Переменная x имеет null-значение. После проверки условия в операторе «if x < 2 then A else B; C» будет выполняться оператор:

 0 A

 1 B

 0 C

1. Переменная x имеет null-значение. После проверки условия в операторе «if x = null then A else B; С»: будет выполняться оператор:

 0 A

 1 B

 0 C

1. Значение выражения «IfNull('ABC' + null, 'abc')»:

 0 0

 0 1

 0 null

 0 false

 0 true

 0 ошибка

 1 'abc'

 0 'ABC'

0 'ABCnull'

1. В табличной форме представления отношений (в чисто реляционной модели):

0 заголовок многоярусный

 1 заголовок одноярусный

 1 имена столбцов уникальны

 0 имена столбцов неуникальны

 0 порядок перечисления столбцов существенен

 1 порядок перечисления столбцов несущественен

 0 порядок строк существенен

 1 порядок строк несущественен

 0 дубликаты строк допустимы

 1 дубликаты строк не допустимы

 1 тело таблицы - это множество строк

 0 тело таблицы - это мультимножество строк

 1 данные в столбце должны иметь простой тип

 0 данные в столбце могут быть составными объектами

 1 значения в столбце должны иметь один и тот же тип

 0 значения в столбце могут иметь различный тип

1. Способы задания домена:

 1 перечислением значений

 0 графическим способом

 1 характеристическим предикатом

 1 порождающей процедурой

 [0]0 указанием типа данных

 0 указанием флажка допустимости null-значений

1. Синоним понятия «атрибут»:

 1 заголовок столбца

 0 строка заголовков столбцов

 0 ячейка тела таблицы

 0 строка тела таблицы

 0 таблица (ее тело)

 0 множество именованных схем таблиц

 0 множество таблиц

1. Синоним понятия «схема отношения»:

 0 заголовок столбца

 1 строка заголовков столбцов

 0 ячейка тела таблицы

 0 строка тела таблицы

 0 таблица (ее тело)

 0 множество именованных схем таблиц

 0 множество таблиц

1. Понятие именованного значения атрибута определяется в терминах:

 1 имя атрибута

 1 домен атрибута

 0 множество атрибутов

 1 значение атрибута

0 множество именованных значений атрибутов

 0 область определения кортежа

 0 множество кортежей

 0 множество именованных схем отношений

 0 множество отношений

 [0]0 флажок допустимости null-значений

 0 множество имен атрибутов

1. Понятие кортежа определяется в терминах:

 [0]0 имя атрибута

 [0]0 домен атрибута

 0 множество атрибутов

 [0]0 значение атрибута

 1 множество именованных значений атрибутов

 1 область определения кортежа

 0 множество кортежей

 0 множество именованных схем отношений

 0 множество отношений

 [0]0 флажок допустимости null-значений

 0 множество имен атрибутов

1. Понятие отношения определяется в терминах:

 [0]0 имя атрибута

 [0]0 домен атрибута

 0 множество атрибутов

 [0]0 значение атрибута

 [0]0 множество именованных значений атрибутов

 [0]0 область определения кортежа

 1 множество кортежей

 0 множество именованных схем отношений

 0 множество отношений

 [0]0 флажок допустимости null-значений

 0 множество имен атрибутов

1. Понятие схемы базы данных определяется в терминах:

 [0]0 имя атрибута

 [0]0 домен атрибута

 [0]0 множество атрибутов

 [0]0 значение атрибута

 0 множество именованных значений атрибутов

 0 область определения кортежа

 0 множество кортежей

 1 множество именованных схем отношений

 0 множество отношений

 [0]0 флажок допустимости null-значений

 [0]0 множество имен атрибутов

1. Понятие базы данных определяется в терминах:

 [0]0 имя атрибута

 [0]0 домен атрибута

 [0]0 множество атрибутов

 [0]0 значение атрибута

 [0]0 множество именованных значений атрибутов

 [0]0 область определения кортежа

 [0]0 множество кортежей

 [0]0 множество именованных схем отношений

 1 множество отношений

 [0]0 флажок допустимости null-значений

 [0]0 множество имен атрибутов

1. Реляционная база данных - это:

 0 хранилище графической информации

 1 набор данных, расположенных в таблицах

 0 набор команд

 0 таблица для обработки логической информации

1. Основная структура в реляционной модели данных:

 0 разность

 0 сумма

 0 произведение

 1 отношение

1. Множество кортежей – это:

 0 домен

 1 отношение

 0 атрибут

 0 журнал

1. Операция выборки – это:

 0 выбор столбцов с определенными номерами

 0 выбор столбцов с определенными наименованиями

 0 выбор строк с определенными номерами

 1 выбор строк, удовлетворяющих условию

 0 выборочное соединение строк таблиц

1. Операция выборки, ее свойства:

 1 идемпотентность

 1 монотонность

 [0]0 коммутативность

 [0]0 ассоциативность

1. Операция проекции – это:

 0 выбор столбцов с определенными номерами

 1 выбор столбцов с определенными наименованиями

 0 выбор строк с определенными номерами

 0 выбор строк, удовлетворяющих условию

 0 выборочное соединение строк таблиц

1. Операция проекции, ее свойства:

 1 идемпотентность

 1 монотонность

 0 коммутативность

 [0]0 ассоциативность

1. Операции объединения и пересечения, их свойства:

 1 идемпотентность

 0 монотонность

 1 коммутативность

 1 ассоциативность

1. Операция декартова произведения, ее свойства:

 0 идемпотентность

 0 монотонность

 1 коммутативность

 1 ассоциативность

1. При реализации какой операции реляционной алгебры в операторе select используется необязательная опция distinct:

 0 выборка

 1 проекция

 0 переименование атрибутов

 0 объединение

 0 разность

 0 декартово произведение

1. При реализации какой операции реляционной алгебры в операторе select используется необязательная опция all:

 0 выборка

 0 проекция

 0 переименование атрибутов

 1 объединение

 0 разность

 0 декартово произведение

1. Для реализации какой операции реляционной алгебры в операторе select используется операция cross join:

 0 выборка

 0 проекция

 0 переименование атрибутов

 0 объединение

 0 разность

 1 декартово произведение

1. Типы данных с объявляемой точностью:

 0 целый

 1 вещественный

 1 десятичный с фиксированной точкой

 0 денежный

 0 логический

 0 строки бит

 0 строки символов

 0 даты и времени

1. Тип счетчика может быть использован:

 0 в программном коде

 1 при объявлении ключа

 0 как счетчик числа строк в таблице

1. BLOB – это:

 0 упорядоченный тип данных

 1 название типов данных

1. Объявление первичного ключа гарантирует

 1 ограничение уникальности

 0 неизбыточность ограничения

 1 определенность атрибутов

1. Объявление первичного ключа не препятствует

 0 вставке кортежей

 0 обновлению кортежей

 1 удалению кортежей

1. Индексы могут

 1 быть простыми и составными

 1 быть уникальными и неуникальными

 0 создаваться только для пустых отношений

1. Объявление виртуального атрибута задает

 0 тип

 0 ограничение значений

 0 флажок допустимости null-значений

 1 формулу

 0 значение по умолчанию

1. Объявление ограничения кортежа использует

 1 имена атрибутов

 0 имена ссылочных отношений

 0 имена атрибутов ссылочных отношений

 1 операции отрицания, конъюнкции, дизъюнкции

 0 кванторы общности и существования

1. Объявление первичного ключа использует имена

 1 атрибутов

 0 ссылочных отношений

 0 атрибутов ссылочных отношений

1. Объявление кандидатного ключа использует имена

 1 атрибутов

 0 ссылочных отношений

 0 атрибутов ссылочных отношений

1. Объявление внешнего ключа использует имена

 1 атрибутов

 1 ссылочных отношений

 1 атрибутов ссылочных отношений

1. Целостность базы данных – это:

 0 защита от несанкционированного доступа

 1 защита от ошибочных изменений

 0 надежность хранения

1. К уровню атрибута относится ограничение

 1 null-значения

 0 кортежа

 0 простого первичного ключа

 0 составного первичного ключа

 0 ссылочной целостности

1. К уровню кортежа относится ограничение

 0 null-значения

 1 кортежа

 0 простого первичного ключа

 0 составного первичного ключа

 0 ссылочной целостности

1. К уровню отношения относится ограничение

 0 null-значения

 0 кортежа

 1 простого первичного ключа

 1 составного первичного ключа

 0 ссылочной целостности

1. К уровню базы данных относится ограничение

 0 null-значения

 0 кортежа

 0 простого первичного ключа

 0 составного первичного ключа

 1 ссылочной целостности

1. Утверждения о правилах поддержания ссылочной целостности:

 0 правило ограничения применимо лишь в случае, когда внешний ключ не допускает null-значений

 0 правило каскадного обновления применимо и в случае, когда внешний ключ является суррогатным (типа счетчика)

1 правило каскадного удаления применимо и в случае, когда внешний ключ является суррогатным (типа счетчика)

 0 правило присвоения null-значений применимо и в случае, когда внешний ключ является и первичным

1. Наиболее близко понятию транзакции утверждение, что это:

 0 корректный доступ к данным

 1 SQL-код, поддерживающий целостность базы данных

 0 трансляция программы

 0 протокол работы

1. Утверждения о триггерах:

 0 триггеры в отличие от хранимых процедур вызываются явно

 1 триггеры в отличие от хранимых процедур вызываются автоматически

 1 триггеры используются для контроля целостности данных

 0 триггеры не могут привести к появлению «мертвых» блокировок

1. Утверждения о представлениях:

 0 все представления являются обновляемыми

 1 некоторые представления являются обновляемыми

 0 все представления являются материализованными

 1 некоторые представления являются материализованными

1. Нормализация актуальна для

 1 базовых отношений OLTP-систем

 0 виртуальных отношений OLTP-систем

 0 хранилищ данных OLAP-систем

1. Ограничение функциональной зависимости может быть

 0 на уровне атрибута

 0 на уровне кортежа

 1 на уровне отношения

 1 навязано объявлением ключей

1. Ограничение функциональной зависимости не препятствует

 0 вставке кортежа в отношение

 0 обновлению кортежа в отношении

 1 удалению кортежа из отношения

1. Подсистему независимых образуют правила

 0 тривиальности

 1 псевдотранзитивности

 1 рефлексивности

 0 аддитивности

 1 пополнения

 0 проективности

1. В рефлексивной функциональной зависимости

 0 левая часть – строгое подмножество правой

 0 левая часть – строгое надмножеством правой

 1 левая и правая части совпадают

 0 левая и правая части не пересекаются

1. В тривиальной функциональной зависимости

 0 левая часть – строгое подмножество правой

 1 левая часть – строгое надмножеством правой

 0 левая и правая части совпадают

 0 левая и правая части не пересекаются

1. В полностью нетривиальной функциональной зависимости

 0 левая часть – строгое подмножество правой

 0 левая часть – строгое надмножеством правой

 0 левая и правая части совпадают

 1 левая и правая части не пересекаются

1. Отношение в 1NF может иметь

 1 простые однозначные атрибуты

 0 составные или многозначные атрибуты

 0 атрибуты с многозначной семантикой

 1 зависимости ключевых атрибутов не только от ключей

 1 зависимости неключевых атрибутов не только от ключей

1. Ключевой атрибут – это атрибут

 1 первичного ключа

 1 кандидатного ключа

 0 внешнего ключа

1. Неполная функциональная зависимость – это зависимость

 0 от части внешних ключей

 1 от части первичного ключа

 1 от части кандидатного ключа

 0 от части внешнего ключа

1. Отношение в 2NF может иметь

 1 простые однозначные атрибуты

 0 составные или многозначные атрибуты

 0 атрибуты с многозначной семантикой

 1 зависимости ключевых атрибутов не только от ключей

 1 зависимости неключевых атрибутов не только от ключей

1. Отношение в 3NF может иметь

 1 простые однозначные атрибуты

 0 составные или многозначные атрибуты

 0 атрибуты с многозначной семантикой

 1 зависимости ключевых атрибутов не только от ключей

 0 зависимости неключевых атрибутов не только от ключей

1. Отношение в NFBC может иметь

 1 простые однозначные атрибуты

 0 составные или многозначные атрибуты

 0 атрибуты с многозначной семантикой

 0 зависимости ключевых атрибутов не только от ключей

 0 зависимости неключевых атрибутов не только от ключей

1. Декомпозицией можно получить независимые отношения в

 1 1NF

 1 2NF

 1 3NF

 0 NFBC

1. Нормальные формы вложены в порядке

 0 NFBC, 1NF, 2NF, 3NF

 0 1NF, NFBC, 2NF, 3NF

 0 1NF, 2NF, NFBC, 3NF

 1 1NF, 2NF, 3NF, NFBC

1. Модель логического уровня, ее элементы:

 1 классы сущностей

 1 атрибуты

 1 связи

 0 СУБД

1. Связь, ее характеристики:

 1 наименование роли

 1 кратность роли

 1 наименование связи

 0 кратность связи

 1 тип связи

1. Диаграмма презентационная, представляет:

 0 все классы сущностей и связи

 0 все первичные ключи

 0 все внешние ключи

 0 все атрибуты

 1 связи многие-ко-многим

 1 составные атрибуты

 1 многозначные атрибуты

 0 ссылки на кандидатные ключи

1. Диаграмма ключевая, представляет:

 1 все классы сущностей и связи

 1 все первичные ключи

 1 все внешние ключи

 0 все атрибуты

 0 связи многие-ко-многим

 1 составные атрибуты

 0 многозначные атрибуты

 0 ссылки на кандидатные ключи

1. Диаграмма полная атрибутивная, представляет:

 1 все классы сущностей и связи

 1 все первичные ключи

 1 все внешние ключи

 1 все атрибуты

 0 связи многие-ко-многим

 0 составные атрибуты

 0 многозначные атрибуты

 1 ссылки на кандидатные ключи

1. ВСЕГДА ли атрибут с маркером:

 0 PK является первичным ключом

 0 FK является внешним ключом

1. ВСЕГДА ли атрибуты с маркерами

 1 PK и PF образуют первичный ключ

 0 PF и FK образуют внешний ключ

1. Связи, обязательные на родительском конце:

 1 идентифицирующая полностью

 1 идентифицирующая неполностью

 1 неидентифицирующая обязательная

 1 неидентифицирующая необязательная

1. Связи, обязательные на дочернем конце:

 0 идентифицирующая полностью

 0 идентифицирующая неполностью

 0 неидентифицирующая обязательная

 0 неидентифицирующая необязательная

1. Иерархическая рекурсия, вид связей реализации:

 0 идентифицирующая полностью

 0 идентифицирующая неполностью

 0 неидентифицирующая обязательная

 1 неидентифицирующая необязательная

1. Иерархическая рекурсия, классы реализации:

 0 дуги

 0 категориальные сущности

 0 обобщенные сущности

 1 ребра

 0 узлов иерархия

 0 узлы

 0 узлы доли

 0 экземпляры агрегатов

 0 экземпляры композитов

 0 экземпляры компонентов

1. Иерархическая рекурсия и ее обобщения – это:

 1 взвешенное дерево

 1 лес

 0 граф

 0 двудольный мультиграф

 1 дерево

 0 мультиграф

 1 иерархия с двумя предками

 0 взвешенный граф

 0 двудольный граф

1. Сетевая рекурсия, вид связей реализации:

 0 идентифицирующая полностью

 1 идентифицирующая неполностью

 0 неидентифицирующая обязательная

 0 неидентифицирующая необязательная

1. Сетевая рекурсия, классы реализации:

 1 дуги

 0 категориальные сущности

 0 обобщенные сущности

 0 ребра

 0 узлов иерархия

 1 узлы

 0 узлы доли

 0 экземпляры агрегатов

 0 экземпляры композитов

 0 экземпляры компонентов

1. Сетевая реализация иерархической рекурсии, вид связей реализации:

 1 идентифицирующая полностью

 0 идентифицирующая неполностью

 1 неидентифицирующая обязательная

 0 неидентифицирующая необязательная

1. Сетевая реализация иерархической рекурсии, классы реализации:

 1 дуги

 0 категориальные сущности

 0 обобщенные сущности

 0 ребра

 0 узлов иерархия

 1 узлы

 0 узлы доли

 0 экземпляры агрегатов

 0 экземпляры композитов

 0 экземпляры компонентов

1. Сетевая рекурсия и ее обобщения – это:

 1 мультиграф

 1 взвешенный граф

 0 лес

 0 иерархия с двумя предками

 0 двудольный мультиграф

 0 взвешенный граф

 0 двудольный граф

 0 дерево

 1 граф

1. Ассоциация, вид связей реализации:

 1 идентифицирующая полностью

 1 идентифицирующая неполностью

 1 неидентифицирующая обязательная

 1 неидентифицирующая необязательная

1. Детализация связей многие-ко-многим, классы реализации:

0 дуги

 0 категориальные сущности

 0 обобщенные сущности

 1 ребра

 0 узлов иерархия

 0 узлы

 1 узлы доли

 0 экземпляры агрегатов

 0 экземпляры композитов

 0 экземпляры компонентов

1. Ассоциация и ее обобщения – это:

 0 иерархия с двумя предками

 0 взвешенное дерево

 0 лес

 0 взвешенный граф

 1 двудольный граф

 0 граф

 0 дерево

 1 двудольный мультиграф

 0 мультиграф

1. Обобщение, вид связей реализации:

 1 идентифицирующая полностью

 0 идентифицирующая неполностью

 0 неидентифицирующая обязательная

 0 неидентифицирующая необязательная

1. Обобщение, классы реализации:

 0 дуги

 1 категориальные сущности

 1 обобщенные сущности

 0 ребра

 0 узлов иерархия

 0 узлы

 0 узлы доли

 0 экземпляры агрегатов

 0 экземпляры композитов

 0 экземпляры компонентов

1. Обобщение, символ:

 0 треугольник закрашенный

 1 треугольник незакрашенный

 0 прямоугольник закрашенный

 0 прямоугольник незакрашенный

 0 ромб закрашенный

 0 ромб незакрашенный

1. Композиция, вид связей реализации:

 1 идентифицирующая полностью

 1 идентифицирующая неполностью

 1 неидентифицирующая обязательная

 0 неидентифицирующая необязательная

1. Обобщение, классы реализации:

 0 дуги

 0 категориальные сущности

 0 обобщенные сущности

 0 ребра

 0 узлов иерархия

 0 узлы

 0 узлы доли

 0 экземпляры агрегатов

 1 экземпляры композитов

 1 экземпляры компонентов

1. Композиция, символ:

 0 треугольник закрашенный

 0 треугольник незакрашенный

 0 прямоугольник закрашенный

 0 прямоугольник незакрашенный

 1 ромб закрашенный

 0 ромб незакрашенный

1. Агрегация, вид связей реализации:

 0 идентифицирующая полностью

 0 идентифицирующая неполностью

 0 неидентифицирующая обязательная

 1 неидентифицирующая необязательная

1. Агрегация, классы реализации:

 0 дуги

 0 категориальные сущности

 0 обобщенные сущности

 0 ребра

 0 узлов иерархия

 0 узлы

 0 узлы доли

 1 экземпляры агрегатов

 0 экземпляры композитов

 1 экземпляры компонентов

1. Агрегация, символ:

 0 треугольник закрашенный

 0 треугольник незакрашенный

 0 прямоугольник закрашенный

 0 прямоугольник незакрашенный

 0 ромб закрашенный

 1 ромб незакрашенный

1. Агрегация общего вида, символ:

 0 треугольник закрашенный

 0 треугольник незакрашенный

 0 прямоугольник закрашенный

 0 прямоугольник незакрашенный

 1 ромб закрашенный

 1 ромб незакрашенный

1. Унификация атрибутов – это:

 0 соглашения об именовании атрибутов

 0 выбор регистра при именовании атрибутов

 1 слияние атрибутов при миграции

**Задания для лабораторных занятий**

### Лабораторные работы проводятся для получения навыков проектирования , программирования и использования баз данных в СУБД [pgAdmin](https://www.google.ru/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&sqi=2&ved=0ahUKEwjz4JaMvf_OAhWqHJoKHcuKCJkQFggcMAA&url=https%3A%2F%2Fwww.pgadmin.org%2F&usg=AFQjCNGQgbeyDM8Pkle0thdm11vIRmYFMg&sig2=u8LZT1qQ8gGfuD0vg5sDGA&bvm=bv.131783435,d.bGs).

### Консультации, выдача лабораторных заданий и прием результатов выполнения осуществляется только во время аудиторных занятий. Задания выполняются последовательно. Правильное выполнение некоторых заданий возможно только, если студент корректно выполнил предыдущие задания. Поэтому приступать к следующему заданию студент может, только сдав преподавателю результат выполнения предыдущего. Результаты выполнения лабораторных работ демонстрируются преподавателю. Во время приема выполненной работы преподаватель вправе:

###  1) Требовать у студента демонстрации выполнения программного проекта, предусмотренной заданием.

### 2) Самостоятельно производить манипуляции с программным проектом и средой программирования, не изменяя программы, составленной студентом.

###  3) Требовать у студента пояснений, относящихся к исходному коду и способам реализации программы.

.

|  |  |
| --- | --- |
| Балл | Критерий оценки |
| 16-20 | работа выполнена полностью и правильно, сделаны правильные выводы; студент четко и последовательно отвечает на поставленные вопросы преподавателя |
| 11-15 | работа выполнена правильно с учетом 2-3 несущественных ошибок исправленных самостоятельно по требованию преподавателя. |
| 6-10 | работа выполнена правильно не менее чем на половину или допущена существенная ошибка |
| 1-5 | допущены существенные ошибки в ходе работы, которые студент не может исправить даже по требованию преподавателя. |
| 0 | работа не выполнена. |

Лабораторная работа 1. Создание и заполнение баз данных

1. Построить базу данных поставщики-товары. Содержащую информацию о поставщиках, товарах и том, какие детали какой поставщик предлагает (предусмотреть таблицу связи). При создании базы данных следует учесть, что название поставщиков и название товара не может быть не определено. Цена товара и его вес всегда имеют положительные значения. При удалении товаров из таблицы товаров соответствующие строки в таблице связи также удаляются, а при удалении поставщика товар в таблице связи остается.

Инструкция SQL позволяющая создать таблицу:

**CREATE TABLE имя\_таблицы (**

 **Название\_поля1 домен атрибутов поля/** **ограниценая,**

 **Название\_поля2 домен атрибутов поля/** **ограниценая,**

 **Название\_поля3 домен атрибутов поля/** **ограниценая,**

 **Название\_поля4 домен атрибутов поля/** **ограниценая**

**);**

Для создания таблицы хранящей информацию о поставщиках, используется следующая SQL инструкция:

  CREATE TABLE SUPPLIER

 (SNO   INTEGER PRIMARY KEY,

 SNAME VARCHAR(20) NOT NULL,

CITY  VARCHAR(20)

);

Здесь:

SNO INTEGER PRIMARY KEY, задает ограничение первичного ключа на столбец с именем SNO.

SNAME VARCHAR(20) NOT NULL, задает ограничение NOT NULL на содержимое столбца с именем SNAME

Ограничение целостности check (.) позволяет задать для определённой колонки, выражение, которое будет осуществлять проверку, помещаемого в эту колонку значения:

CREATE TABLE products (

 product\_no integer PRIMARY KEY,

 name text,

 price numeric CHECK (price > 0));

Ограничение внешнего ключа:

CREATE TABLE orders (

 order\_id integer PRIMARY KEY,

 product\_no integer REFERENCES products (product\_no),

 quantity integer

);

Здесь product\_no integer REFERENCES products (product\_no) показано, что таблица orders является *ссылающейся (потомком)* таблицей, а таблица products является *ссылочной (предком)* таблицей.

Синтаксически задание внешнего ключа c ограничением на удаление CASCAD имеет вид:

имя\_столбца домен REFERENCES имя\_таблицы\_предка (имя\_столбца\_таблицы\_предка) ON DELETE CASCAD

Инструкция позволяющая удалить таблицу:

**DROP TABLE название\_таблицы;**

2. Заполните полученную базу данных и проконтролируйте ее заполнение.

Для помещения записей в таблицу используется оператор INSERT:

***INSERT INTO название\_таблицы VALUES (название\_атрибута\_поля1, название\_атрибута\_поля2,..);***

Пример: ***NSERT INTO*** SUPPLIER ***VALUES (1, 'Smith', 'London');***

Запросе, выдающий всю таблицу целиком

***SELECT \* FROM название\_таблицы;***

Вопросы по лабораторной работе 1:

1. Что такое СУБД?
2. Реляционные базы данных это?
3. Что такое домен атрибута?
4. Первичный ключ это?
5. Как организуется отношение предок / потомок в реляционных базах данных?
6. Средства для обеспечения поддержания целостности базы данных?
7. В чем особенность null – значений?

Лабораторная работа 2. **Модификация структуры баз данных и обновление информации.**

Для обновления данных используется инструкция

*UPDATE название\_таблицы SET название\_колонки = новое значение WHERE условие отбора*

Для удаления строк, используйте инструкцию [DELETE](file:///C%3A%5CC%3A%5C%D0%A0%D1%9A%D0%A0%D1%95%D0%A0%D1%91%20Web%20%D0%A0%D0%8E%D0%A0%C2%B0%D0%A0%E2%84%96%D0%A1%E2%80%9A%D0%A1%E2%80%B9%5Chttp___postgresql.ru.net_manual_sql.html%5Cpostgresql.ru.net%5Cmanual%5Csql-delete.html) :

*DELETE FROM название\_таблицы WHERE условие отбора;*

Чтобы добавить колонку, используйте команду вида:

*ALTER TABLE название\_таблицы ADD COLUMN название\_колонки домен\_колонки;*

Чтобы удалить колонку, используйте команду вида:

*ALTER TABLE название\_таблицы DROP COLUMN название\_колонки;*

Чтобы переименовать колонку, используйте команду вида:

*ALTER TABLE name\_table RENAME COLUMN name\_column TO new\_name\_column*

Чтобы добавить какое-либо ограничение на таблицу, используется табличный синтаксис определения этого ограничения. Например:

*ALTER TABLE products ADD CHECK (name <> '');*

*ALTER TABLE products ADD CONSTRAINT some\_name UNIQUE (product\_no);*

*ALTER TABLE products ADD FOREIGN KEY (product\_group\_id) REFERENCES product\_groups;*

*ALTER TABLE name\_table ALTER COLUMN name\_column SET NOT NULL;*

*Чтобы удалить ограничение, вам необходимо знать его имя:*

*ALTER TABLE название\_таблицы DROP CONSTRAINT some\_name;*

*ALTER TABLE название\_таблицы ALTER COLUMN название\_столбцы DROP NOT NULL:*

Чтобы переименовать таблицу, используйте команду вида:

*ALTER TABLE name\_table RENAME TO new\_name\_table;*

Чтобы преобразовать колонку к другому типу данных, используйте команду вида:

*ALTER TABLE название\_таблицы ALTER COLUMN название\_столбца TYPE новый\_домен;*

Задания к лабораторной работе 2:

1. *Обновить значение цены и цвета товара с указанным именем.*
2. *Изменить имя поставщика из указанного города*
3. *Удалите из таблицы товаров товар с указанным именем. Проконтролируйте выполнение операции. Посмотрите как при удалении товара изменилось состояние таблицы связи. Ответе на вопрос, почему так получилось?*
4. *. Добавьте в БД поставщика из Москвы по имени Alex (можно предложить свое имя) поставляющего детали Screw и Cam.*
5. *. А затем удалите его из списка поставщиков, проконтролировав при этом состояние таблицы связи.*
6. *Добавьте в таблицу поставщиков колонки для хранения телефонов и контактного лица. Заполните колонки*
7. *Удалите из таблицы поставщиков колонку содержащую контактные лица.*
8. *Переименуйте колонку, содержащую телефон по своему желанию. Внесите в БД имения таким образом, чтобы в ней не могло быть поставщиков с неопределенными телефонами.*
9. *Измените домен атрибута цвет*

 Лабораторная работа 3. **Запросы на выборку и итоговые операторы, запросы с группировкой, подзапросы**

Для получения данных из какой-либо таблицы, к этой таблице осуществляется *запрос*. Для этого используется оператор SQL SELECT.

SELECT /*часть, где перечисляются возвращаемые запросом поля*/

  FROM *table\_name\_1 / перечисляется из каких таблиц/*

   WHERE / перечисляются *условия выбора строк*

*GROUP BY / перечисляются столбцы группировки*

*HAVING / перечисляются условия отбора групп*

*ORDER BY /указываются столбцы по которым пойдет сортировка и направление сортировки*

**Итоговые операторы**

SQL снабжён итоговыми операторами (например, AVG, COUNT, SUM, MIN, MAX), которые принимают название атрибута в качестве аргумента. Значение итогового оператора высчитывается из всех значений заданного атрибута (столбца) всей таблицы. Если в запросе указана группа, то вычисления выполняются только над значениями группы.

**Запросы с группировкой**

Следующий запрос позволяет выяснить, сколько типов товаров продает каждый из поставщиков

SELECT S.SNO, S.SNAME, COUNT(SE.PNO)

  FROM SUPPLIER S, SELLS SE

  WHERE S.SNO = SE.SNO

  GROUP BY S.SNO, S.SNAME;

**Пример Вложенная выборка**

Если мы хотим узнать все детали, имеющие цену больше чем деталь 'Screw', то используем запрос:

**SELECT** \*   **FROM** PART

  **WHERE** PRICE > (**SELECT** PRICE **FROM** PART   **WHERE** PNAME='Screw');

Задания для лабораторной работы 3

1. *Получить все кортежи из таблицы Товаров, где цена больше 10*
2. *Получите только цену и название товаров, цена которых с 15% наценкой меньше 15.*
3. *Предоставте информацию о название и цвете товара, где название товара bolt или цена товара меньше 15.*
4. *Постройте запрос, выдающий название деталей и их стоимость с 30% наценкой, при этом переименуйте столбец price. Результирующая таблица должна содержать только кортежи, где цена с наценкой не ниже 15.*
5. *Построить запрос, где цена детали попадает в интервал [10,20] расположите строки по убыванию цены детали.*
6. *Построить запрос, выдающий имена поставщиков начинающихся с буквы ‘B’*.
7. *Построить запрос, выдающий поставщиков располагающихся в городах Лондон или Париж в алфавитном порядке.*
8. *Постройте запрос, позволяющий узнать количество типов деталей, хранящихся в таблице PART.*
9. *Каков средний вес деталей?*
10. *Покажите минимальную, максимальную и среднюю стоимости продаваемых деталей.*
11. *Сколько имеется поставщиков из Парижа?*
12. *Какова суммарная стоимость красных деталей?*
13. *Постройте запрос выдающий поставщиков, продающих более одной детали.*
14. *Постройте запрос выдающий поставщика продающего деталь с минимальной стоимостью*
15. *Постройте запрос выдающий детали предлагаемые несколькими поставщиками*
16. *Постройте запрос выдающий все детали продаваемые Адамсем.*
17. *Постройте запрос так, чтобы узнать какие поставщики продают деталь Bolt.*
18. *Запрос позволяющий получить название и телефон поставщиков не поставляющих деталь Nut.*
19. *Запрос позволяющий получить название поставщиков поставляющих хотя бы одну деталь Cam или зеленую деталь.*
20. *Постройте запрос так, чтобы узнать какие детали поставляются из Лондона и кем*

Лабораторная работа 4. **Объединение таблиц, многотабличные запросы. Представления**.

Создадим простую базу данных из двух таблиц: таблица

мальчиков и таблица девочек. Таблицы содержат имена и города проживания.

1.Создать декартово произведение можно следующим образом:

|  |  |
| --- | --- |
| Select \*From Boys, Girls | Select \*From Boys CROSS JOIN Girls |

2. Создать внутреннее объединение можно также двумя способами:

|  |  |
| --- | --- |
| Select \*From Boys, GirlsWhere Boys.City=Gilse.City | Select \*From Boys **INNER JOIN** Girls**ON** Boys.City=Gilse.City |

3. Внешнее левое внешнее объединение

|  |
| --- |
| Select \*From Boys **LEFT OUTER JOIN** Girls**ON** Boys.City=Gilse.City |

**Представления**

Для определения представлений в SQL используется команда **CREATE VIEW**. Синтаксис:

  CREATE VIEW *view\_name*  AS *select\_stmt*

где *select\_stmt*, допустимое выражение выборки. Заметим, что *select\_stmt* не выполняется при создании представления. Оно только сохраняется в *системных каталогах* и выполняется всякий раз когда делается запрос представления.

Задание для лабораторной работы 4

1. *Создайте внешнее объединение таблицы поставщиков и таблице связи*
2. *Постройте запрос, объединяющий 3 таблицы по их общим атрибутам, и выдающий в результат имя поставщика и название детали.*
3. *Выполните операцию правого соединения таблиц поставщиков и связи, выдавая в результат номера деталей, номера и имена поставщиков*
4. *Постройте запрос выдающий, сколько поставщиков продают деталь с указанным названием.*
5. *Какова суммарная стоимость деталей поставляемых поставщиком, имя которого начинается с указанной буквы Создайте представление возвращающее объединение трех таблиц.*
6. *Постройте запрос так, чтобы узнать какие поставщики продают деталь с указанным названием.*
7. *Получите поставщиков из указанного города и названия предлагаемых ими деталей*
8. *Есть ли поставщики из Лондона продающие детали с ценой меньше 30.*
9. *С помощью представления выясните* ***сколько*** *поставщиков продают деталь с указанным названием.*

**Промежуточная аттестация**

**Методические указания.** Промежуточная аттестация по дисциплине «Базы данных» проводится в виде устного экзамена. Учебным планом по направлению подготовки «01.03.02 Прикладная математика и информатика» предусмотрена одна промежуточная аттестация по соответствующим разделам данной дисциплины. Подготовка студента к прохождению промежуточной аттестации осуществляется в период лекционных и практических занятий, а также во внеаудиторные часы в рамках самостоятельной работы. Во время самостоятельной подготовки студент пользуется конспектами лекций, основной и дополнительной литературой по дисциплине (см. перечень литературы в рабочей программе дисциплины).

***Критерии оценивания.*** Во время экзамена студент должен дать развернутый ответ на вопросы, изложенные в билете. Преподаватель вправе задавать дополнительные вопросы по всему изучаемому курсу.

Во время ответа студент должен продемонстрировать знания по формализации решения прикладных задач c использованием технологии баз данных; разработке требований к созданию и сопровождению БД и ее компонентов; реализации проектных решений с использованием современных информационно-коммуникационных технологий; внедрению проектов автоматизации решения прикладных задач; сопровождению и использованию баз данных в информационных системах. Полнота ответа определяется показателями оценивания планируемых результатов обучения (раздел 2).

**Список вопросов к устному экзамену:**

1. Эволюция систем баз данных.
2. Типы баз данных.
3. Данные мультимедиа.
4. Интеграция информации.
5. Обзор структуры и технологий СУБД.
6. Реляционная алгебра. Формализация понятий.
7. Неопределенные значения, интерпретации и свойства.
8. Основные унарные и бинарные операции. Свойства операций.
9. Внутреннее соединение.
10. Левое, правое и полное внешние соединения. Их основное свойство.
11. Реляционное исчисление и полнота реляционной алгебры.
12. Базовая структура оператора select.
13. Выражение операций реляционной алгебры.
14. Целостность при переходах.
15. Триггеры.
16. Внедрение SQL-операторов в прикладные программы.
17. Базовые типы данных.
18. Пользовательские типы данных.
19. Ключи. Ограничение уникальности. Простые и составные ключи.
20. Суперключи. Первичные, кандидатные, внешние ключи.
21. Индексы.
22. Структура базовых отношений.
23. Базовые атрибуты. Виртуальные атрибуты.
24. Схемы базовых отношений.
25. Операторы создания, модификации и удаления базовых отношений.
26. Целостность по состоянию.
27. Ограничения уровней атрибута, кортежа, отношения, базы данных.
28. Правила поддержания ссылочной целостности.
29. Понятие функциональной зависимости.
30. Правила вывода Армстронга.
31. Производные правила вывода.
32. Независимость и полнота системы правил Армстронга.
33. Ключи и навязывание функциональных зависимостей.
34. Простые/составные и однозначные/многозначные атрибуты.
35. Ключевые атрибуты. Полная функциональная зависимость.
36. 1НФ, 2НФ, 3НФ, НФБК. Вложенность нормальных форм.
37. Уровни детализации ER-диаграмм.
38. Миграция ключей. Типы связей.
39. Иерархическая рекурсия.
40. Сетевая рекурсия.
41. Ассоциация.
42. Обобщение.
43. Композиция.
44. Агрегация.
45. Унификация атрибутов.
46. Управление транзакциями.

ФОС для проведения текущего контроля и промежуточной аттестации одобрен на заседании кафедры математического и компьютерного моделирования (протокол № \_\_\_\_ от \_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_\_\_года).

Автор:

к.ф-м.н., доцент О.М. Ромакина