Федеральное государственное бюджетное образовательное

учреждение высшего образования

Московский государственный университет имени М.В. Ломоносова

Факультет вычислительной математики и кибернетики

**УТВЕРЖДАЮ**

декан факультета вычислительной математики и кибернетики

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_/И.А. Соколов /**

**«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_\_г.**

**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ**

**Наименование дисциплины:**

**Кафедральный практикум (СП)**

**Уровень высшего образования:**

**бакалавриат**

**Направление подготовки / специальность:**

**01.03.02 «Прикладная математика и информатика» (3++)**

**Направленность (профиль):**

**Системное программирование и компьютерные науки**

**Форма обучения:**

**очная**

**Москва 2023**

Рабочая программа дисциплины (модуля) разработана в соответствии с самостоятельно установленным МГУ образовательным стандартом (ОС МГУ) для реализуемых основных профессиональных образовательных программ высшего образования по направлению подготовки 01.03.02, 01.04.02 "Прикладная математика и информатика" программы бакалавриата Утвержден приказом МГУ от 30 августа 2019 года № 1041 (в редакции приказов МГУ от 11 сентября 2019 года № 1109, от 10 июня 2021 года № 609, от 7 октября 2021 года № 1048, от 21 декабря 2021 года № 1404, от 2 ноября 2022 года № 1299)

**1.** Дисциплина относится к вариативной части ОПОП ВО.

**2.** Входные требования для освоения дисциплины (модуля): учащиеся должны владеть знаниями по алгоритмам, алгоритмическим языкам и программированию в объеме, соответствующем программе первых трёх лет обучения основных образовательных программ бакалавриата по укрупненным группам направлений и специальностей 01.00.00 «Математика и механика», 02.00.00 «Компьютерные и информационные науки».

**3.** Результаты обучения по дисциплине (модулю), соотнесенные с требуемыми компетенциями выпускников*.*

Компетенции выпускников, частично формируемые при реализации дисциплины (модуля):

* **ПК-4.Б** Способность применять в профессиональной деятельности современные средства разработки программного обеспечения и методы параллельной обработки данных, реализовывать отдельные этапы разработки системного и прикладного программного обеспечения, систем обработки и анализа данных, сетевых технологий.

Планируемые результаты обучения по дисциплине (модулю):

**Знать:**

1. базовые средства языка SQL, включая описания схемы базы данных, запросов для выборки, удаления, изменения и добавления данных;
2. расширенные возможности языка SQL, включая средства контроля целостности с помощью транзакций и триггеров, средства оптимизации, средства работы с представлениями и функциями, средства управления доступом.

**Уметь:**

1. проектировать схемы реляционных баз данных, в том числе денормализованные, при помощи ER-диаграмм, языка DDL;
2. манипулировать данными при помощи запросов SELECT, INSERT, UPDATE, DELETE;
3. определять ограничения целостности и обеспечивать их выполнение при помощи механизма транзакций и триггеров;
4. анализировать длительность выполнения запросов и оптимизировать реляционную базу данных для сокращения этой длительности;
5. реализовывать представления и функции в реляционной базе данных;
6. управлять доступом пользователей к базе данных.

**Владеть:**

1. навыками проектирования и реализации реляционных баз данных в СУБД PostgreSQL и среде разработки pgAdmin.

**4.** Формат обучения: семинарские занятия проводятся с использованием компьютерного класса.

**5.** Объем дисциплины (модуля) составляет 2 з.е., в том числе 26 академических часов, отведенных на контактную работу обучающихся с преподавателем, 46 академических часов на самостоятельную работу обучающихся.

**6.** Содержание дисциплины (модуля), структурированное по темам (разделам) с указанием отведенного на них количества академических часов и виды учебных занятий.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Наименование и краткое содержание разделов и тем дисциплины (модуля),**  **Форма промежуточной аттестации по дисциплине (модулю)** | **Всего**  **(часы**) | В том числе | | | |
| **Контактная работа  Виды контактной работы, часы** | | | **Самостоятельная работа**  **обучающегося,**  **часы** |
| Занятия  лекционного типа\* | Занятия  семинарского типа\* | **Всего** |
| 1. Введение в SQL*(простейшие запросы для выборки данных).* | **4** | 0 | 2 | **2** | **2** |
| 1. Создание реляционной базы данных *(проектирование и описание схемы БД, наполнение БД)*. | **4** | 0 | 2 | **2** | **2** |
| 1. Текущий контроль успеваемости: сдача практических заданий №№1, 2.1, 2.2. | **6** | 0 | 2 | **2** | **4** |
| 1. Манипулирование данными и контроль их целостности средствами языка SQL *(операторы SELECT, DELETE, UPDATE, INSERT; описания ограничений целостности; использование механизма транзакций и триггеров)*. | **4** | 0 | 2 | **2** | **2** |
| 1. Текущий контроль успеваемости: сдача практических заданий   №№ 2.3, 2.4. | **6** | 0 | 2 | **2** | **4** |
| 1. Дополнительные средства проектирования схем баз данных   *(денормализация, использование JSON)* | **4** | 0 | 2 | **2** | **2** |
| 1. Оптимизация выполнения запросов *(получение значений показателей выполнения запросов, оптимизация с помощью индексов, оценка эффективности выполнения оптимизированных запросов)*. | **6** | 0 | 2 | **2** | **4** |
| 1. Текущий контроль успеваемости: сдача практических заданий №№ 3.1, 3.2. | **8** | 0 | 4 | **4** | **4** |
| 1. Дополнительные средства СУБД для аналитики: представления и функции *(упрощение манипулирования данными с помощью представлений и функций).* | **6** | 0 | 2 | **2** | **4** |
| 1. Средства управления доступом к данным *(операции с пользовательскими аккаунтами и правами доступа)*. | **6** | 0 | 2 | **2** | **4** |
| 1. Текущий контроль успеваемости: сдача практических заданий №№ 3.3, 3.4. | **8** | 0 | 4 | **4** | **4** |
| Промежуточная аттестация: зачет | **10** | 0 | 0 | **0** | **10** |
| **Итого** | **72** | **0** | **26** | **26** | **46** |

**7. Фонд оценочных средств (ФОС)для оценивания результатов обучения по дисциплине (модулю)**

**7.1. Типовые варианты практических заданий и вопросов при их сдаче для проведения текущего контроля успеваемости**.

7.1.1. Практическое задание №1

Первое практическое задание заключается в знакомстве со средой pgAdmin и написании простейших SQL-запросов с использованием оператора SELECT. Для модельной базы данных «KingCorporation» должны быть составлены 4 запроса согласно индивидуальному варианту. Скрипт для создания и заполнения модельной базы данных и её описание: http://sp.cs.msu.ru/prak3/sample\_db.sql

После составления запросов следует убедиться в их правильности при помощи более простых запросов.

Темы для проработки в рамках задания:

Основные понятия реляционных и SQL-ориентированных баз данных.

• Синтаксис SQL https://postgrespro.ru/docs/postgrespro/11/sql-syntax

• SELECT-запросы https://postgrespro.ru/docs/postgrespro/11/queries

• Функции и операторы https://postgrespro.ru/docs/postgrespro/11/functions

• Оконные функции https://postgrespro.ru/docs/postgrespro/9.5/tutorial-window

• Полное описание синтаксиса встретившихся команд https://postgrespro.ru/docs/postgrespro/11/sql-commands

• Структура учебной базы данных «KingCorporation»

• Работа в среде pgAdmin

Примеры вопросов по заданию

• Объяснить, как работают написанные запросы.

• Рассказать про операцию соединения (JOIN) и различные её разновидности.

• Рассказать про агрегатные функции, предложения GROUP BY и HAVING.

• Как выбрать только уникальные значения какого-либо столбца?

• Как осуществить сортировку по возрастанию/убыванию по значению какого-либо столбца?

• Как агрегатные функции ведут себя по отношению к неопределённым значениям?

• Рассказать о теоретико-множественных операциях в SQL.

• Чем отличаются UNION и UNION ALL?

• Чем отличаются COUNT(\*) и COUNT(field)?

• Как подсчитать количество уникальных значений столбца?

• Как можно осуществить проверку на неопределенное значение?

• Рассказать про предикат LIKE.

• Как можно выбрать только определенное количество строк?

• Чем SQL-таблица отличается от отношения?

• Исправить неверно работающий запрос (запросы).

• Упростить один или несколько запросов.

• Округлить результирующее значение до 3 знаков после точки.

• Округлить вещественное число до целого без нулей после точки.

• Переписать запрос, не используя функцию MAX (MIN).

• Изменить формат вывода данных (например, формат даты и времени).

• Написать или модифицировать запрос по сформулированному заданию.

7.1.2. Блок практических заданий №№ 2.1-2.4

Блок практических заданий 2.1-2.4 призван сформировать у студента понимание особенностей хранения данных приложения в РСУБД и умение настраивать и поддерживатьэто хранение.

*Практическое задание №2.1 Проектирование схемы базы данных*

Задание связано с проектированием схемы базы данных для работы приложения (WEB/Mobile/Desktop). Каждый индивидуальный вариант содержит предметную область, из которой должна быть проектируемая база данных. Задачей студента является решить, для чего будет использоваться создаваемая база данных, и, исходя из этого, построить её концептуальную схему. Результатом выполнения задания является схема базы данных (в виде ER-диаграммы, содержащей таблицы и связи между ними, без уточнения типов столбцов). При сдаче задания студент должен обосновать соответствие созданной схемы поставленной задаче.

Для проектирования схемы и построения диаграммы можно использовать любые средства, один из вариантов использовать сайт:

https://www.lucidchart.com/pages/examples/er-diagram-tool

*Типовые индивидуальные варианты (сквозные, т. к. используются в блоках практических заданий №№2.1-2.4, 3.1-3.4):*

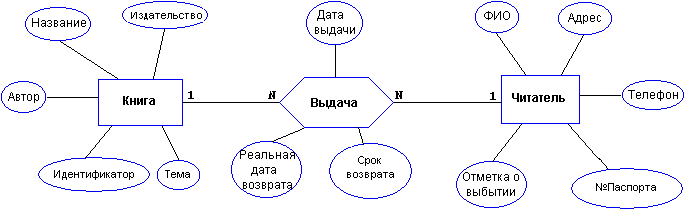
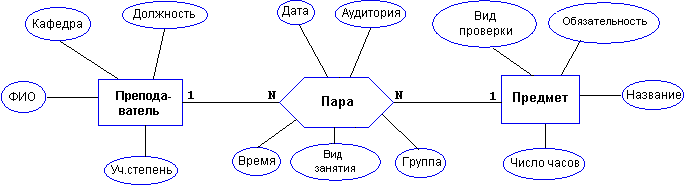
Вариант №1. Библиотека

Рис. Пример схемы базы данных для варианта №1

Список запросов для реализации на языке SQL:

1. Выбрать значения задержки поставки товара после заказа для всех покупателей, живущих в штате 'CA' (Калифорния)
2. Выбрать названия и города тех отделов, у которых нет начальников (должность - 'MANAGER')
3. Выбрать даты и имена покупателей тех заказов, которые обслуживал продавец 'SMITH' и в которых сумма заказа превышала кредит покупателя.
4. Выбрать названия продукта, для которого в 1991 г. было продано минимальное количество экземпляров продукта.

## Вариант №2. Университет

Рис. Пример схемы базы данных для варианта №2

Список запросов для реализации на языке SQL:

1. Выбрать цены на сегодняшний день всех продуктов, в названии которых есть слова 'WIFF SOFTBALL'.
2. Выбрать названия и города отделов, в которых есть аналитики, и их количество (должность - 'ANALYST').
3. Выбрать названия и города тех отделов, в которых есть сотрудники, не являющиеся продавцами (должность - не 'SALESPERSON'), но получающие комиссионные.
4. Выбрать минимальную сумму продаж, которая приходится на одного сотрудника, работающего в городе 'NEW YORK'.

*Темы для проработки в рамках задания*

• Модель "сущность-связь" (ER-модель).

• Первичные и внешние ключи.

• Типы связей и их моделирование.

• Нормальные формы и нормализация.

*Требования к созданной схеме базы данных*

• Схема должна соответствовать поставленной задаче.

• Связи между сущностями должны быть правильно смоделированы.

• Таблицы должны удовлетворять, по крайней мере, третьей нормальной форме.

• Желательно придерживаться какой-либо системы в именовании таблиц и столбцов.

*Практическое задание №2.2 Создание и заполнение таблиц*

Задание заключается в подготовке SQL-скрипта для создания таблиц согласно схеме, созданнойпри выполнении предыдущего задания (с уточнением типов столбцов). Необходимо определить первичные и внешние ключи, а также декларативные ограничения целостности (возможность принимать неопределенное значение, уникальные ключи, проверочные ограничения и т. д.). Таблицы следует создавать в отдельной базе данных. Кроме того, нужно подготовить данные для заполнения созданных таблиц. Объем подготовленных данных должен составлять не менее 10 экземпляров для каждой из стержневых сущностей и 20 экземпляров для каждой из ассоциативных. На основе этих данных необходимо создать SQL-скрипт для вставки соответствующих строк в таблицы БД.

*Темы для проработки в рамках задания*

• Язык DDL, операторы CREATE TABLE и ALTER TABLE https://postgrespro.ru/docs/postgrespro/11/ddl-basics

https://postgrespro.ru/docs/postgrespro/11/ddl-default

https://postgrespro.ru/docs/postgrespro/11/ddl-alter

• Типы данных https://postgrespro.ru/docs/postgrespro/11/datatype

• Декларативные ограничения целостности https://postgrespro.ru/docs/postgrespro/11/ddl-constraints

• Оператор INSERT https://postgrespro.ru/docs/postgrespro/11/dml-insert https://postgrespro.ru/docs/postgrespro/11/dml-returning

• Полное описание синтаксиса встретившихся команд https://postgrespro.ru/docs/postgrespro/11/sql-commands

*Примеры вопросов при сдаче задания*

• Объяснить, что делают написанные запросы.

• В чём различие типов CHAR и VARCHAR? VARCHAR и TEXT?

• Что такое внешний ключ?

• Какие существуют способы поддержания ссылочной целостности?

• Что такое уникальный ключ?

• Что такое SERIAL?

• Рассказать о значениях по умолчанию и неопределенных значениях.

• Как можно хранить даты и время?

• Рассказать о числовых типах данных.

• Каким образом можно вставить несколько строк с помощью одного оператора INSERT?

• Как ведет себя оператор INSERT, если в списке столбцов перечислены не все столбцы?

• Добавить какие-либо ограничения целостности.

• Добавить SERIAL.

• Исправить выявленные при проверке недочеты.

*Практическое задание №2.3. Операторы манипулирования*

Задание посвящено манипулированию данными с помощью операторов SQL. В ходе выполнения задания необходимо:

• Подготовить еще 3-4 выборки, которые имеют осмысленное значение для предметной области, и также составить для них SQL-скрипты.

• Сформулировать 3-4 запроса на изменение и удаление из базы данных

Запросы должны быть сформулированы в терминах предметной области. Среди запросов обязательно должны быть такие, которые будут вызывать срабатывание ограничений целостности. Составить SQL-скрипты для выполнения этих запросов.

*Темы для проработки в рамках задания:*

• Оператор SELECT https://postgrespro.ru/docs/postgrespro/11/queries

• Оператор UPDATE https://postgrespro.ru/docs/postgrespro/11/dml-update

• Оператор DELETE https://postgrespro.ru/docs/postgrespro/11/dml-delete

• Декларативные ограничения целостности https://postgrespro.ru/docs/postgrespro/11/ddl-constraints

• Полное описание синтаксиса встретившихся команд https://postgrespro.ru/docs/postgrespro/11/sql-commands

*Примеры вопросов при сдаче задания:*

• Объяснить, как работают написанные запросы.

• Примеры вопросов по оператору SELECT см. в практическом задании №1.

• Исправить неверно работающий запрос (запросы).

• Упростить один или несколько запросов.

• Написать или модифицировать запрос по сформулированному заданию.

*Практическое задание №2.4 Контроль целостности данных*

Задание посвящено контролю целостности данных, который производится с помощью механизма транзакций и триггеров. Транзакции позволяют рассматривать группу операций как единое целое, либо отрабатывают все операции, либо ни одной. Это позволяет избегать несогласованности данных. Триггеры позволяют проверять целостность данных в момент выполнения транзакций, поддерживать целостность, внося изменения, и откатывать транзакции, приводящие к потере целостности.

Необходимо подготовить SQL-скрипты для проверки наличия аномалий (потерянных изменений, грязных чтений, неповторяющихся чтений, фантомов) при параллельном исполнении транзакций на различных уровнях изолированности SQL/92 (READ UNCOMMITTED, READ COMMITTED, REPEATABLE READ, SERIALIZABLE). Подготовленные скрипты должны работать с одной из таблиц, созданных в практическом задании №2.1. Для проверки наличия аномалий потребуются два параллельных сеанса, операторы в которых выполняются пошагово:

• Установить в обоих сеансах уровень изоляции READ UNCOMMITTED. Выполнить сценарии проверки наличия аномалий потерянных изменений и грязных чтений.

• Установить в обоих сеансах уровень изоляции READ COMMITTED. Выполнить сценарии проверки наличия аномалий грязных чтений и неповторяющихся чтений.

• Установить в обоих сеансах уровень изоляции REPEATABLE READ. Выполнить сценарии проверки наличия аномалий неповторяющихся чтений и фантомов.

• Установить в обоих сеансах уровень изоляции SERIALIZABLE. Выполнить сценарий проверки наличия фантомов.

Необходимо составить скрипт для создания триггера, а также подготовить несколько запросов для проверки и демонстрации его полезных свойств:

• Изменение данных для сохранения целостности.

• Проверка транзакций и их откат в случае нарушения целостности.

*Темы для проработки в рамках задания:*

• Понятие транзакции, свойства транзакций. https://postgrespro.ru/docs/postgrespro/11/tutorial-transactions

• Уровни изолированности и аномалии https://postgrespro.ru/docs/postgrespro/11/transaction-iso

• Триггеры и триггерные функции https://postgrespro.ru/docs/postgrespro/11/trigger-definition

https://postgrespro.ru/docs/postgrespro/11/plpgsql-trigger

• Сообщения и ошибки https://postgrespro.ru/docs/postgrespro/11/plpgsql-errors-and-messages

• Полное описание синтаксиса встретившихся команд https://postgrespro.ru/docs/postgrespro/11/sql-commands

*Примеры вопросов при сдаче задания:*

• Рассказать об аномалиях доступа к БД.

• Перечислить аномалии, возникающие на каждом из уровней изолированности.

• Рассказать о свойствах транзакций.

• Рассказать об управлении транзакциями.

• Что такое тупики? Как бороться с тупиками?

• На каком уровне изолированности возможны тупики?

• Как обеспечивается изолированность транзакций в СУБД?

• Как бороться с проблемой фантомов?

• Что такое журнал транзакций?

• Как обеспечивается постоянство хранения (durability) в СУБД?

• Объяснить принцип работы написанного триггера.

• Какие бывают типы триггеров?

• Когда может срабатывать триггер?

• В каком порядке срабатывают триггеры?

• Можно ли менять порядок срабатывания триггеров?

• Сработает ли триггер, если оператор, выполненный пользователем, не затрагивает ни одну строку таблицы?

• Продемонстрировать откат транзакции при возникновении ошибок.

• Продемонстрировать возникновение тупика.

• Исправить неверные сценарии проверки аномалий

• Исправить ошибки в работе триггера.

• Модифицировать триггер каким-либо образом.

7.1.3. Блок практических заданий №№ 3.1-3.4

Блок практических заданий №№ 3.1-3.4 призван сформировать у студента понимание особенностей создания аналитических баз данных и умение их настраивать и поддерживать.

*Практическое задание №3.1 Проектирование схемы базы данных*

Задание связано с проектированием схемы базы данных для аналитики. Будем исходить из того, что приложение, для которого была сделана база данных в задании №2.1, стало очень популярным и по нему каждый день можно собирать большой объем статистической информации. Что это будет за статистика? Почему именно ее необходимо собирать, обрабатывать и анализировать? Задачей студента является ответить на эти вопросы, и, исходя из этого, разработать базу данных и заполнить ее данными. Результатом данного практического задания является схема базы данных, скрипты создания базы данных и ее заполнения, обладающие следующими свойствами:

• Как минимум одна таблица должна содержать не меньше 100 млн. записей, которые со временем теряют актуальность.

• Другая таблица, связанная с первой, должна содержать не меньше 1 млн. записей.

• В одной из таблиц с количество записей больше 1 млн. должна быть колонка с текстом, по которой будет необходимо настроить полнотекстовый поиск.

• В одной из таблиц с количество записей больше 1 млн. должна быть колонка с данными в json-формате.

• В одной из таблиц с количество записей больше 1 млн. должна быть колонка с массивом.

При выполнении задания важно учитывать плюсы и минусы денормализации схемы данных и использования массивов и json-формата. При сдаче задания студент должен обосновать соответствие созданной схемы поставленной задаче.

Для проектирования схемы и построения диаграммы можно использовать любые средства, один из вариантов использовать сайт:

https://www.lucidchart.com/pages/examples/er-diagram-tool

*Темы для проработки в рамках задания:*

• Денормализация https://habr.com/ru/company/latera/blog/281262/ https://habr.com/ru/post/64524/

• Массивы https://postgrespro.ru/docs/postgrespro/11/arrays https://postgrespro.ru/docs/postgrespro/11/functions-array

• JSON https://postgrespro.ru/docs/postgrespro/11/datatype-json https://postgrespro.ru/docs/postgrespro/11/functions-json

• Наполнение базы данных https://postgrespro.ru/docs/postgrespro/11/populate

*Практическое задание №3.2 Оптимизация*

Задание посвящено ускорению выполнения запросов. Для этого могут быть использованы механизмы секционирования, наследования и индексов. Для выполнения задания необходим достаточно большой объем данных, чтобы оптимизация была целесообразной (порядка 1 млн. строк в каждой таблице). Необходимо подготовить два запроса:

• Запрос к одной таблице, содержащий фильтрацию по нескольким полям.

• Запрос к нескольким связанным таблицам, содержащий фильтрацию по нескольким полям.

Для каждого из этих запросов необходимо провести следующие шаги:

• Получить план выполнения запроса без использования индексов.

• Получить статистику (IO и Time) выполнения запроса без использования индексов.

• Создать нужные индексы, позволяющие ускорить запрос.

• Получить план выполнения запроса с использованием индексов и сравнить с первоначальным планом.

• Получить статистику выполнения запроса с использованием индексов и сравнить с первоначальной статистикой.

• Оценить эффективность выполнения оптимизированного запроса.

Также необходимо продемонстрировать полезность индексов для организации полнотекстового поиска, фильтрации с использованием массива и json-формата. Для таблицы объемом больше 100 млн. записей произвести оптимизацию, позволяющую быстро удалять старые данные, ускорить вставку и чтение данных.

*Темы для проработки в рамках задания:*

• EXPLAIN https://postgrespro.ru/docs/postgrespro/11/using-explain https://postgrespro.ru/docs/postgrespro/11/planner-stats

https://postgrespro.ru/docs/postgrespro/11/explicit-joins

• ANALYZE https://postgrespro.ru/docs/postgrespro/11/routine-vacuuming#VACUUM-FOR-STATISTICS

• Индексы https://postgrespro.ru/docs/postgrespro/11/indexes

• Полнотекстовый поиск https://postgrespro.ru/docs/postgrespro/11/textsearch

• Наследование таблиц https://postgrespro.ru/docs/postgrespro/11/ddl-inherit

• Секционирование таблиц https://postgrespro.ru/docs/postgrespro/11/ddl-partitioning

• Полное описание синтаксиса встретившихся команд https://postgrespro.ru/docs/postgrespro/11/sql-commands

*Примеры вопросов при сдаче задания:*

• В чем отличие первичного ключа и уникального индекса?

• В каких случаях имеет смысл создавать индексы? Какие колонки следует включать в индекс и почему?

• Какие существуют способы внутренней организации индексов?

• Рассказать о проблеме фрагментации индексов. Как бороться с фрагментацией?

• Имеет ли значение порядок указания колонок при создании индекса?

• В чем разница между IndexScan и IndexSeek?

• В чем разница между секционированием и наследованием?

• Зачем нужен ANALYZE?

• Исправить ошибки в подготовленных выборках.

• Могут ли индексы ухудшить производительность? Если да, то продемонстрировать это.

• На что влияет порядок сортировки (ASC\DESC) при создании индекса? Продемонстрировать это.

• Продемонстрировать полезность индекса по выражению.

• Продемонстрировать полезность частичного индекса.

*Практическое задание №3.3 Представления и функции*

Задание посвящено упрощению работы аналитика и ограничения его возможностей с помощью созданию и использованию представлений и функций: требуется составить SQL-скрипты для создания нескольких представлений и функций, которые позволяли бы упростить манипуляции с данными или позволяли бы не показывать аналитику всю базу данных, предоставив только необходимую часть информации.

*Темы для проработки в рамках задания:*

• Представления https://postgrespro.ru/docs/postgrespro/11/sql-createview https://postgrespro.ru/docs/postgrespro/11/rules-views

https://postgrespro.ru/docs/postgrespro/11/rules-materializedviews

• Функции https://postgrespro.ru/docs/postgrespro/11/xfunc-sql

• Полное описание синтаксиса встретившихся команд https://postgrespro.ru/docs/postgrespro/11/sql-commands

*Примеры вопросов при сдаче задания:*

• Объяснить, как работают написанные запросы.

• Рассказать о CHECK OPTION.

• Рассказать о модификации данных через представления.

• Рассказать о вставке данных через представления.

• Примеры вопросов по оператору SELECT см. в задании №1.

• Исправить неверно работающий запрос (запросы).

• Упростить один или несколько запросов.

• Продемонстрировать изменение и вставку данных через представления.

• Написать или модифицировать запрос по сформулированному заданию.

• Продемонстрировать полезность материализованного представления.

*Практическое задание №3.4 Управление доступом*

В рамках выполнения задания происходит освоение способов управления доступом. При выполнении задания необходимо:

• Создать пользователя test и выдать ему доступ к базе данных.

• Составить и выполнить скрипты присвоения новому пользователю прав доступа к таблицам, созданным в практическом задании №3.1. При этом права доступа к различным таблицам должны быть различными, а именно:

o По крайней мере, для одной таблицы новому пользователю присваиваются права SELECT, INSERT, UPDATE в полном объеме.

o По крайней мере, для одной таблицы новому пользователю присваиваются права SELECT и UPDATE только избранных столбцов.

o По крайней мере, для одной таблицы новому пользователю присваивается только право SELECT.

• Присвоить новому пользователю право доступа (SELECT) к представлению, созданному в практическом задании №3.1

• Создать стандартную роль уровня базы данных, присвоить ей право доступа (UPDATE на некоторые столбцы) к представлению, созданному в практическом задании №3.3, назначить новому пользователю созданную роль.

• Выполнить от имени нового пользователя некоторые выборки из таблиц и представления. Убедиться в правильности контроля прав доступа.

• Выполнить от имени нового пользователя операторы изменения таблиц с ограниченными правами доступа. Убедиться в правильности контроля прав доступа.

*Темы для проработки в рамках задания:*

• Роли и пользователи https://postgrespro.ru/docs/postgresql/11/user-manag

https://postgrespro.ru/docs/postgrespro/11/app-createuser

• Директивы GRANT и REVOKE https://postgrespro.ru/docs/postgrespro/11/ddl-priv

https://postgrespro.ru/docs/postgrespro/11/ddl-schemas#DDL-SCHEMAS-PRIV

• Полное описание синтаксиса встретившихся команд https://postgrespro.ru/docs/postgrespro/11/sql-commands

*Примеры вопросов при сдаче задания:*

• Для чего нужны роли?

• Что такое схема?

• Рассказать про директивы GRANT и REVOKE.

• Для чего нужна роль PUBLIC?

• Как добавить нового пользователя в текущую базу данных?

• Как позволить пользователю заходить на сервер?

• Какие существуют права?

• Исправить ошибки в обязательной части.

• Сменить владельца базы данных.

• Сменить пароль для пользователя.

• Определить роль с заданными правами.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ШКАЛА И КРИТЕРИИ ОЦЕНИВАНИЯ результатов обучения (РО) по дисциплине (модулю)** | | | | |
| Оценка  РО и соответствующие  виды оценочных средств | 2 | 3 | 4 | 5 |
| **Знания** о базовых средствах языка SQL, включая описания схемы базы данных, запросов для выборки, удаления, изменения и добавления данных; о расширенных возможностях языка SQL, включая средства контроля целостности с помощью транзакций и триггеров, средства оптимизации, средства работы с представлениями и функциями, средства управления доступом (вопросы при сдаче практических заданий) | *Отсутствие знаний* обазовых средствах языка SQL, включая описания схемы базы данных, запросов для выборки, удаления, изменения и добавления данных; о расширенных возможностях языка SQL, включая средства контроля целостности с помощью транзакций и триггеров, средства оптимизации, средства работы с представлениями и функциями, средства управления доступом | *Фрагментарные знания* обазовых средствах языка SQL, включая описания схемы базы данных, запросов для выборки, удаления, изменения и добавления данных; о расширенных возможностях языка SQL, включая средства контроля целостности с помощью транзакций и триггеров, средства оптимизации, средства работы с представлениями и функциями, средства управления доступом | *Общие, но не структурированные знания* обазовых средствах языка SQL, включая описания схемы базы данных, запросов для выборки, удаления, изменения и добавления данных; о расширенных возможностях языка SQL, включая средства контроля целостности с помощью транзакций и триггеров, средства оптимизации, средства работы с представлениями и функциями, средства управления доступом | Сформированные систематические знания обазовых средствах языка SQL, включая описания схемы базы данных, запросов для выборки, удаления, изменения и добавления данных; о расширенных возможностях языка SQL, включая средства контроля целостности с помощью транзакций и триггеров, средства оптимизации, средства работы с представлениями и функциями, средства управления доступом |
| **Умения** проектировать схе-мы реляционных баз данных, в том числе денормализованные, при помощи ER-диаграмм, языка DDL;манипулировать данными при помощи запросов SELECT,INSERT,UPDATE,DELETE;определять ограничения целостности и обеспечивать их выполнение при помощи механизма транзакций и триггеров;анализировать длительность выполнения запросов и оптимизировать реляционную базу данных для сокращения этой длительности;реализовывать представления и функции в реляционной базе данных;управлять доступом пользователей к базе данных(практические задания) | *Отсутствие умений*проектировать схемы реляционных баз данных, в том числе денормализованные, при помощи ER-диаграмм, языка DDL;манипулировать данными при помощи запросов SELECT,IN-SERT,UPDATE,DELETE;определять ограничения целостности и обеспечивать их выполнение при помощи механизма транзакций и триггеров;анализировать длительность выполнения запросов и оптимизировать реляционную базу данных для сокращения этой длительности;реализовывать представления и функции в реляционной базе данных;управлять доступом пользователей к базе данных | *В целом успешное, но не систематическое умение*проектировать схемы реляционных баз данных, в том числе денормализованные, при помощи ER-диаграмм, языка DDL;манипулировать данными при помощи запросов SELECT,IN-SERT,UPDATE,DELETE;определять ограничения целостности и обеспечивать их выполнение при помощи механизма транзакций и триггеров;анализировать длительность выполнения запросов и оптимизировать реляционную базу данных для сокращения этой длительности;реализовывать представления и функции в реляционной базе данных;управлять доступом пользователей к базе данных | *В целом успешное, но с пробелами и неточностями непринципиального характера умение*проектировать схе-мы реляционных баз данных, в том числе денормализованные, при помощи ER-диаграмм, языка DDL;манипулировать данными при помощи запросов SELECT,INSERT,UPDATE,DELETE;определять ограничения целостности и обеспечивать их выполнение при помощи механизма транзакций и триггеров;анализировать длительность выполнения запросов и оптимизировать реляционную базу данных для сокращения этой длительности;реализовывать представления и функции в реляционной базе данных;управлять доступом пользователей к базе данных | *Успешное и систематическое умение*проектировать схе-мы реляционных баз данных, в том числе денормализованные, при помощи ER-диаграмм, языка DDL;манипулировать данными при помощи запросов SELECT,INSERT,UPDA-TE,DELETE;определять ограничения целостности и обеспечивать их выполнение при помощи механизма транзакций и триггеров;анализировать длительность выполнения запросов и оптимизировать реляционную базу данных для сокращения этой длительности;реализовывать представления и функции в реляционной базе данных;управлять доступом пользователей к базе данных |
| **Навыки** проектирования и реализации реляционных баз данных в СУБД PostgreSQL и среде разработки pgAdmin (практические задания) | *Отсутствие навыков*проектирования и реализации реляционных баз данных в СУБД PostgreSQL и сре-де разработки pgAdmin | *Наличие отдельных навыков*проектирования и реализации реляционных баз данных в СУБД Post-greSQL и среде разработки pgAdmin | *В целом, сформированные, но используемые не в активной форме навыки*проектирования и реализации реляционных баз данных в СУБД PostgreSQL и среде разработки pgAdmin | *Сформированные, применяемые при решении задач навыки*проектирования и реализации реляционных баз данных в СУБД PostgreSQL и среде разработки pgAdmin |

|  |  |
| --- | --- |
| **Соответствие результатов обучения и компетенций, в развитии которых участвует дисциплина (модуль)** | |
| Результаты обучения | Компетенция, с частичным формированием которой связано достижение результата обучения |
| **Знать:**   1. базовые средства языка SQL, включая описания схемы базы данных, запросов для выборки, удаления, изменения и добавления данных; 2. расширенные возможности языка SQL, включая средства контроля целостности с помощью транзакций и триггеров, средства оптимизации, средства работы с представлениями и функциями, средства управления доступом.   **Уметь:**   1. проектировать схемы реляционных баз данных, в том числе денормализованные, при помощи ER-диаграмм, языка DDL; 2. манипулировать данными при помощи запросов SELECT, INSERT, UPDATE, DELETE; 3. определять ограничения целостности и обеспечивать их выполнение при помощи механизма транзакций и триггеров; 4. анализировать длительность выполнения запросов и оптимизировать реляционную базу данных для сокращения этой длительности; 5. реализовывать представления и функции в реляционной базе данных; 6. управлять доступом пользователей к базе данных.   **Владеть:**  навыками проектирования и реализации реляционных баз данных в СУБД PostgreSQL и среде разработки pgAdmin. | **ПК-4.Б** Способность применять в профессиональной деятельности современные средства разработки программного обеспечения и методы параллельной обработки данных, реализовывать отдельные этапы разработки системного и прикладного программного обеспечения, систем обработки и анализа данных, сетевых технологий. |

8. Ресурсное обеспечение:

Основная литература:

1. Кузнецов С. Д. Базы данных. – М.: Академия, Серия: Университетский учебник,2012 г.
2. Кузнецов С. Д. Базы данных: языки и модели. – М.: Бином, 2008.

Дополнительная литература:

1. Кузнецов С. Д. Основы баз данных. 2-е изд. – М.: Бином. Лаборатория знаний, 2007

Информационныеэлектронные ресурсы:

1. http://sp.cs.msu.ru/prak3/
2. https://www.postgresql.org
3. https://www.pgadmin.org
4. https://postgrespro.ru
5. https://www.lucidchart.com
6. http://citforum.ru/database/advanced\_intro/

Материально-техническое обеспечение:

компьютерный класс с доступом в Internet, используемом при ознакомлении с учебными материалами; специализированное программное обеспечение СУБД PostgreSQL, средаразработки pgAdmin.

9. Язык преподавания - русский.

10. Преподаватели: ассистент Гомзин А. Г., Малахов Д.А., Малышко В.В., Теймуразов К. Б.

11. Авторы программы: ассистент Гомзин А. Г., Малахов Д. А.,Малышко В.В., Теймуразов К. Б.