ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

"ДОНЕЦКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ"

Кафедра программной инженерии

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

К КУРСОВОЙ РАБОТЕ

ПО ДИСЦИПЛИНЕ: «Программирование систем с серверами баз данных»

НА ТЕМУ: «Создание клиент-серверной информационной системы

средствами СУБД»

Руководители:

Щедрин С. В.

Ногтев Е. А.

Выполнил:

ст. гр. ПИ-19б

Носаченко А. А.

Донецк – 2022

# РЕФЕРАТ

Отчет по курсовой работе содержит: \_\_ страницу, \_\_ рисунков, \_ таблицы, \_ приложения, \_ источников.

Объект исследования – клиент-серверная информационная система, созданная средствами системы управления базами данных (далее – СУБД), взаимодействующая с реляционной базой данных (далее – БД) по принципу разделения ролей.

Цель – изучение на практике способов проектирования и реализации реляционных баз данных с применением средств СУБД PostgreSQL, ориентированных на одновременное применение несколькими пользователями, имеющими различные роли, а также разработка соответствующей клиент-серверной программной системы для обеспечения взаимодействия пользователей с БД.

Результат – БД, соответствующая теме задания – «Парикмахерские», серверное приложение управления СУБД, клиентское приложение для взаимодействия с сервером, руководства пользователей.

БД, СУБД, РОЛЬ, ПОЛЬЗОВАТЕЛЬ, СЕРВЕР, КЛИЕНТ, ПАРИКМАХЕРСКАЯ

СОДЕРЖАНИЕ

[РЕФЕРАТ 2](#_Toc106630833)

[СОДЕРЖАНИЕ 3](#_Toc106630834)

[ВВЕДЕНИЕ 5](#_Toc106630835)

[1 ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ, ПОСТАНОВКА ЗАДАЧИ 6](#_Toc106630836)

[2 ОБОСНОВАНИЕ ВЫБОРА СУБД, ОПИСАНИЕ ВОЗМОЖНОСТЕЙ СУБД 8](#_Toc106630837)

[3 ОБОСНОВАНИЕ ВЫБОРА ИНСТРУМЕНТАЛЬНЫХ СРЕДСТВ ДЛЯ НАПИСАНИЯ КЛИЕНТСКОЙ ЧАСТИ, ПРОЕКТИРОВАНИЕ СТРУКТУРЫ ПО 11](#_Toc106630838)

[3.1 Невизуальные компоненты для работы с данными 11](#_Toc106630839)

[3.2 Визуальные компоненты отображения данных 11](#_Toc106630840)

[3.3 Разработка шаблонов приложений для работы c таблицами базы данных 11](#_Toc106630841)

[4 ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ В ВЫБРАННОЙ СУБД 12](#_Toc106630842)

[4.1 Проектирование концептуальной модели БД 12](#_Toc106630843)

[4.2 Создание таблиц, доменов, индексов, сиквенсов 13](#_Toc106630844)

[4.3 Разработка триггеров 13](#_Toc106630845)

[4.4 Организация многоролевого доступа к данным 13](#_Toc106630846)

[4.5 Разграничение доступа к данным на уровне строк 13](#_Toc106630847)

[4.6 Партицирование одной из основных таблиц БД 13](#_Toc106630848)

[4.7 Проектирование запросов к базе данных 14](#_Toc106630849)

[4.8 Создание представлений и хранимых процедур, функций 14](#_Toc106630850)

[5 РАЗРАБОТКА КЛИЕНТСКОГО ПРИЛОЖЕНИЯ 15](#_Toc106630851)

[5.1 Формы и компоненты для работы в роли «Менеджер» 15](#_Toc106630852)

[5.2 Формы и компоненты для работы в роли «Директор» 15](#_Toc106630853)

[5.3 Формы и компоненты для работы в роли «Работник» 15](#_Toc106630854)

[5.4 Генерация результатов не менее трех итоговых запросов (диаграммы, экспорт в Excell) 15](#_Toc106630855)

[6 ТЕСТИРОВАНИЕ РАЗРАБОТАННОЙ ИНФОРМАЦИОННОЙ СИСТЕМЫ 16](#_Toc106630856)

[ВЫВОДЫ 17](#_Toc106630857)

[СПИСОК ЛИТЕРАТУРЫ 18](#_Toc106630858)

[ПРИЛОЖЕНИЕ А. Техническое задание 19](#_Toc106630859)

[ПРИЛОЖЕНИЕ Б. Листинг серверного приложения 25](#_Toc106630860)

[ПРИЛОЖЕНИЕ В. Листинг клиентского приложения 26](#_Toc106630861)

[ПРИЛОЖЕНИЕ Г. Руководство пользователя 27](#_Toc106630862)

[ПРИЛОЖЕНИЕ Д. Руководство администратора 28](#_Toc106630863)

# ВВЕДЕНИЕ

В наше время сложно найти фирму, которая бы не использовала персональные компьютеры для хранения и обработки информации. Одним из лучших способов хранения различной информации является внесение её в базу данных. Всё, с чем мы взаимодействуем в жизни, вероятно, зафиксировано в какой-нибудь базе. Базы данных формируются и работают под управлением специальных программных средств, называемых системами управления базами данных.

База данных – это организованная структура, которая предназначена для хранения информации. В то время, когда происходило развитие термина баз данных, в них сохранялись исключительно информация, однако сейчас многие системы управления базами данных позволяет размещать в своих структурах и данные, и программный код, с помощью которого совершается связь с пользователями или с другими программно-аппаратными комплексами [1]. При этом данные должны не противоречить друг другу, быть целостными и не избыточными. База данных создается для сохранения и непосредственного доступа к информации, содержащей сведения об искомой предметной области.

Система управления базами данных – это программный механизм, предназначенный для записи, поиска, сортировки, обработки и печати информации, содержащейся в базе данных [2].

Накопление хранимого объема информации, рост группы пользователей информационных систем служат источником к обширному развитию комфортных в интерфейсе и относительно лёгких для понимания табличных систем управления базами данных. Создание доступа к информации базы данных сразу нескольких пользователей одновременно, зачастую находящихся на далеком расстоянии от места хранения баз данных, а также друг от друга, в настоящее время является наиболее актуальной проблемой разрабатываемых систем, использующих базы данных. В них решаются проблемы характерные для параллельных процессов, правильности данных, а также получения не санкционированного входа.

# 1 ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ, ПОСТАНОВКА ЗАДАЧИ

В ходе курсовой работы необходимо разработать базу данных с приложения взаимодействия с ней для учета работы парикмахерских города, которая будет содержать информацию о парикмахерских (номер, район города, разряд (высший, первый, второй), тип собственности (частная, государственная, акционерная,…), год начала функционирования, телефон), клиентах (ФИО, дата рождения, социальная группа (предприниматель, банковский служащий, инженер, рабочий,…), домашний адрес)) и оказанных клиентам услугах (парикмахерская , название услуги (стрижка, завивка, укладка, маникюр, массаж,…), стоимость ( зависит от разряда парикмахерской), дата оказания услуги).

Исходя из данной задачи, необходимо выполнить следующие этапы разработки:

1. Спроектировать концептуальную модель базы данных (БД) для предметной области и представить ее в виде взаимосвязанных таблиц, находящихся в третьей нормальной форме. Выделить базовые таблицы и таблицы-справочники, указать для них первичные и внешние ключи.

2. Создать базу данных в среде СУБД средствами языка SQL. Добавить таблицы, домены, индексы.

3. Разработать не менее шести триггеров (по одному для каждого типа события), как минимум для двух различных таблиц БД. Триггеры типа BEFORE INSERT должны быть созданы для всех таблиц и с использованием генераторов задавать значение первичного ключа для вновь добавляемой записи.

4. Заполнить таблицы БД с использованием соответствующих запросов на языке SQL.

5. Сформулировать следующие виды запросов:

- симметричное внутреннее соединение с условием (два запроса с условием отбора по внешнему ключу, два – по датам);

- симметричное внутреннее соединение без условия (три запроса);

- левое внешнее соединение;

- правое внешнее соединение;

- запрос на запросе по принципу левого соединения;

- итоговый запрос без условия;

- итоговый запрос без условия c итоговыми данными вида: «всего», «в том числе»;

- итоговые запросы с условием на данные (по значению, по маске, с использованием индекса, без использования индекса);

- итоговый запрос с условием на группы;

- итоговый запрос с условием на данные и на группы;

- запрос на запросе по принципу итогового запроса;

- запрос с использованием объединения

- запросы с подзапросами.

6. Запросы без параметров реализовать в виде представлений, остальные запросы – в виде хранимых процедур и/или функций. Создать, по меньшей мере, одно модифицируемое представление, используя механизм триггеров. ВСЯ логика проектируемого ПО – на сервере.

7. Разработать клиентское приложение, которое предоставляет следующие возможности для работы с созданной базой данных:

- многопользовательский режим работы (одна программа для всех ролей – ситуативный доступ к интерфейсу)

- наличие нескольких ролей пользователя (менеджер – добавление/удаление/редактирование пользователей, их прав/ролей; директор – просмотр отчётов о прибыли и убытках, работник – создание записей о проводимых работах, изучение личного дохода)

- просмотр содержимого таблиц и представлений (здесь и далее – с учетом прав пользователей);

- добавление, редактирование и удаление записей таблиц и модифицируемых представлений;

- работа с наборами данных, находящимися в отношении «один-ко-многим» (создать составную форму для просмотра и редактирования данных родительской и дочерней таблиц);

- поиск и фильтрация данных отображаемых таблиц;

- просмотр результатов выполнения запросов;

- визуализация результатов одного из итоговых запросов.

8. Обеспечить защиту данных, информации от несанкционированного доступа, сделать защиту на уровне строк, выполнить партицирование одной из основных таблиц.

# 2 ОБОСНОВАНИЕ ВЫБОРА СУБД, ОПИСАНИЕ ВОЗМОЖНОСТЕЙ СУБД

Для разработки системы выбрана СУБД PostgreSQL. Она является одним из нескольких бесплатных популярных вариантов СУБД. Это весьма старая система, поэтому в настоящее время она хорошо развита, и позволяет пользователям управлять как структурированными, так и неструктурированными данными. Может быть использована на большинстве основных платформ, включая Linux (где особенно хорошо проявляется производительность). Прекрасно справляется с задачами импорта информации из других типов баз данных с помощью собственного инструментария. Движок БД может быть размещен в ряде сред, в том числе виртуальных, физических и облачных.

Достоинства СУБД:

- является масштабируемым решением и позволяет обрабатывать терабайты данных;

- поддерживает формат json;

- существует множество предопределенных функций;

- доступен ряд интерфейсов;

- поддержка БД неограниченного размера;

- мощные и надёжные механизмы транзакций и репликации;

- расширяемая система встроенных языков программирования и поддержка загрузки C-совместимых модулей;

- наследование;

- легкая расширяемость.

Идеально подходит для организаций с ограниченным бюджетом, требует привлечения квалифицированных специалистов, когда требуется возможность выбрать уникальный интерфейс и использовать json.

Кроме основных возможностей, присущих любой SQL базе данных, PostgreSQL поддерживает:

- Очень высокий уровень соответствия ANSI SQL 92, ANSI SQL 99 и ANSI SQL 2003.

- Схемы, которые обеспечивают пространство имен на уровне SQL. Схемы содержат таблицы, в них можно определять типы данных, функции и операторы.

- Subqueries – подзапросы (subselects), полная поддержка SQL92. Подзапросы делают язык SQL более гибким и зачастую более эффективным.

- Outer Joins – внешние связки (LEFT,RIGHT, FULL)

- Rules – правила, согласно которым модифицируется исходный запрос.

- Views – представления, виртуальные таблицы. Реальных экземпляров этих таблиц не существуют, они материализуются только при запросе.

- Cursors – курсоры, позволяют уменьшить трафик между клиентом и сервером, а также память на клиенте, если требуется получить не весь результат запроса, а только его часть.

- Table Inheritance – наследование таблиц, позволяющее создавать объекты, которые наследуют структуру родительского объекта и добавлять свои специфические атрибуты.

- Prepared Statements (преподготовленные запросы) – это объекты, живущие на стороне сервера, которые представляют собой оригинальный запрос после команды PREPARE, который уже прошел стадии разбора запроса (parser), модификации запроса (rewriting rules) и создания плана выполнения запроса (planner), в результате чего, можно использовать команду EXECUTE, которая уже не требует прохождения этих стадий. Для сложных запросов это может быть большим выигрышем.

- Stored Procedures – серверные (хранимые) процедуры позволяют реализовывать бизнес логику приложения на стороне сервера. Кроме того, они позволяют сильно уменьшить трафик между клиентом и сервером.

- Savepoints(nested transactions) – в отличие от "плоских транзакций", которые не имеют промежуточных точек фиксации, использование savepoints позволяет отменять работу части транзакции, например в следствии ошибочно введенной команды, без влияния на оставшуюся часть транзакции.

- Права доступа к объектам системы на основе системы привилегий.

- Система обмена сообщениями между процессами – LISTEN и NOTIFY позволяют создать событийную модель взаимодействия между клиентом и сервером.

- Триггеры позволяют управлять реакцией системы на изменение данных, как перед самой операцией (BEFORE), так и после (AFTER).

- Cluster table – упорядочивание записей таблицы на диске согласно индексу, что иногда за счет уменьшения доступа к диску ускоряет выполнение запроса.

Богатый набор типов данных PostgreSQL включает:

- Символьные типы (character(n)) как определено в стандарте SQL и тип text с практически неограниченной длиной.

- Numeric тип поддерживает произвольную точность.

- Массивы согласно стандарту SQL:2003.

- Большие объекты (Large Objects) для бинарных данных размером до 2Gb.

- Геометрические типы для работы с пространственными данными на плоскости.

- ГИС (GIS) типы в PostgreSQL позволяют работать с трехмерными данными.

- Сетевые типы (Network types).

- Композитные типы (composite types).

- Типы времени реализованы с очень большой точностью.

- Псевдотипы serial и bigserial организующие AUTO\_INCREMENT.

Безопасность данных также является важнейшим аспектом любой СУБД. В PostgreSQL она обеспечивается 4-мя уровнями безопасности:

- PostgreSQL нельзя запустить под привилегированным пользователем – системный контекст

- SSL,SSH шифрование трафика между клиентом и сервером – сетевой контекст

- Сложная система аутентификации на уровне хоста или IP адреса/подсети. Система аутентификации поддерживает пароли, шифрованные пароли, Kerberos, IDENT и прочие системы, которые могут подключаться, используя механизм подключаемых аутентификационных модулей.

- Детализированная система прав доступа ко всем объектам базы данных, которая совместно со схемой, обеспечивающая изоляцию названий объектов для каждого пользователя, PostgreSQL предоставляет богатую и гибкую инфраструктуру.

# 3 ОБОСНОВАНИЕ ВЫБОРА ИНСТРУМЕНТАЛЬНЫХ СРЕДСТВ ДЛЯ НАПИСАНИЯ КЛИЕНТСКОЙ ЧАСТИ, ПРОЕКТИРОВАНИЕ СТРУКТУРЫ ПО

## 3.1 Невизуальные компоненты для работы с данными

## 3.2 Визуальные компоненты отображения данных

## 3.3 Разработка шаблонов приложений для работы c таблицами базы данных

# 4 ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ В ВЫБРАННОЙ СУБД

## 4.1 Проектирование концептуальной модели БД

При проектировании концептуальной модели БД основной задачей становится обеспечение корректной взаимосвязи парикмахерских, множества клиентов, посещающих и множества услуг, которые они могут заказать за одно посещение.

Для выполнения данной задачи следует ввести дополнительный объект БД – таблицу, отражающую журнал посещений.

Исходя из этого схема БД, при приведении её к нормальной форме, примет вид, отражённый на рисунке 4.1.

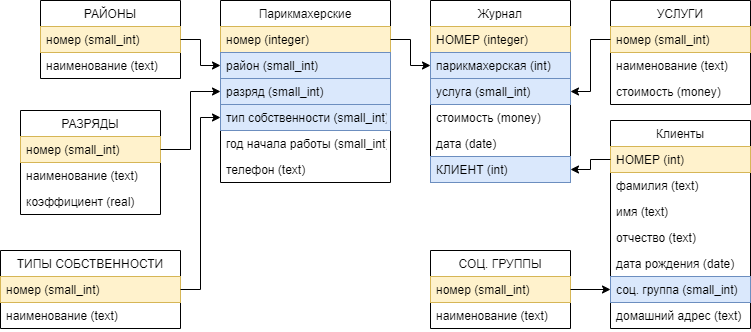


Рисунок 4.1 – Концептуальная модель базы данных «Парикмахерские»

Учитывая ожидаемое количество записей в таблице «Журнал», а также необходимость контроля доступа к данным, данной таблице будет введено внутреннее подразделение по дополнительному полю «Пользователь» при фактической реализации базы.

## 4.2 Создание таблиц, доменов, индексов, сиквенсов

Разрабатываемая БД состоит из трёх основных таблиц, отражающих контролируемые парикмахерские, их клиентов и журналы посещений, а также из пяти редко изменяемых таблиц-списков, к которым не имеют доступа обычные пользователи БД (см. рис. 4.2).

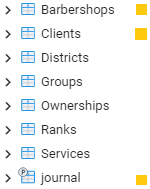


Рисунок 4.2 – Таблицы в БД «Парикмахерские»

Для данной БД не необходимости создавать специализированные домены.

Индексы созданы для каждой из основных таблиц по полям главных ключей, а также дополнительно по фамилии и имени клиента в таблице «Клиенты» и всем внешним ключам таблицы «Журнал» (см. рис. 4.3).

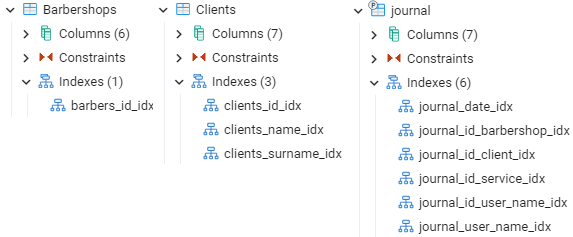


Рисунок 4.3 – Индексы БД

Сиквенсы обеспечивают последовательное повышение числового значения, благодаря чему отлично подходят для реализации идентификации записей таблиц.

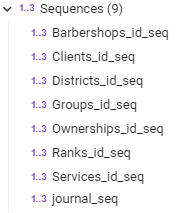


Рисунок 4.4 – Сиквенсы БД

В БД было создано 8 сиквенсов, которые отвечают за последовательную выдачу значений полям «id» в таблицах при срабатывании триггеров перед добавлением записи (см. рис. 4.4).

## 4.3 Разработка триггеров

Исходя из поставленной техническим заданием задачи, для каждой таблицы были созданы простые триггеры перед добавлением, которые, используя соответствующие таблицам сиквенсы, присваивают в поле идентификатора значение, если оно не было задано.

Также созданы следующие триггеры:

- Триггер перед обновлением в таблице «Клиенты» – если происходит изменение в фамилии, имени или отчестве, к соответствующему значению дописывается маркер «(upd)» (см. рис. 4.6 - 4.7).

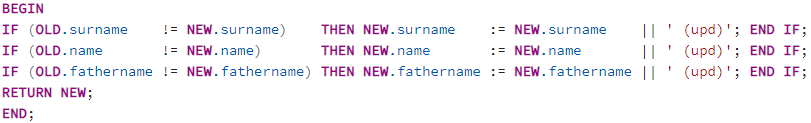


Рисунок 4.6 – Код тела триггера перед обновлением



Рисунок 4.7 – Пример работы триггера перед обновлением

- Триггер перед удалением в таблице «Клиенты» – передаёт все записи в журнале «нулевому» клиенту – записи, созданной для хранения прибыли от клиентов, регистрация которых была удалена (см. рис. 4.8).

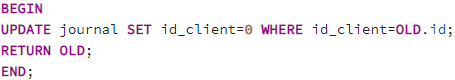


Рисунок 4.8 – Код тела триггера перед удалением

- Триггер после добавления в таблице «Парикмахерские» – если происходит добавление записи о парикмахерской, дата открытия которой позже текущей (то есть её только предстоит открыть), то, независимо от указанных данных, ранг будет удалён (см. рис. 4.9).

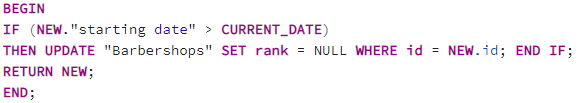


Рисунок 4.9 – Код тела триггера после добавления

- Триггеры после изменения и после удаления в таблице «Парикмахерские» – добавляют новую запись в таблицу «Парикмахерские» с указанными данными до взаимодействия, а также датой взаимодействия, пользователем, который производил взаимодействие и старым (и новым – при взаимодействии изменения) идентификатором записи (см. рис. 4.10 - 4.11).

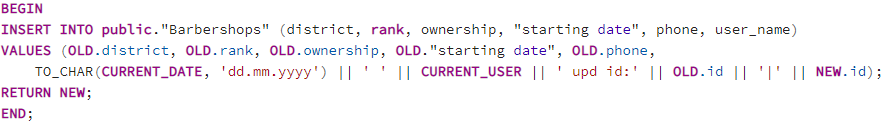


Рисунок 4.10 – Код тела триггера после изменения

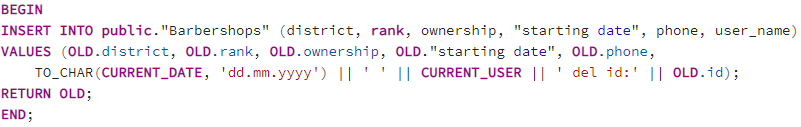


Рисунок 4.11 – Код тела триггера после удаления

Таким образом БД содержит восемь индексных триггеров и пять управляющих, соответствующих заданию.

## 4.4 Организация многоролевого доступа к данным

Для работы с БД было создано три роли-группы пользователей: группа директоров, группа менеджеров и группа работников. Взаимодействие со структурой базы и записями таблиц-справочников доступно только для администратора БД. Для выполнения же работ, группам пользователей предоставлены необходимые и исчерпывающие разрешения на взаимодействие с объектами базы данных посредством созданных хранимых функций (см. пример на рис. 4.5) и представлений таблиц.

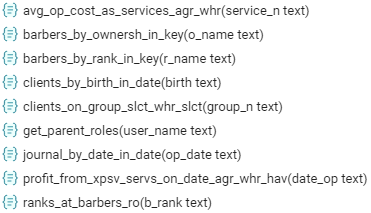


Рисунок 4.5 – Функции БД, созданные на основе запросов,

Указанных в техническом задании

Также следует отметить, что таблица «Журнал» разделена на партиции, доступ к которым имеют лишь пользователи, которым они принадлежат.

## 4.5 Разграничение доступа к данным на уровне строк

Разграничение доступа к данным на уровне строк обеспечивает контроль воздействия пользователей на данные, скрывая записи по определённым ограничениям, что позволяет создать механизм, благодаря которому пользователи смогут влиять только на записи, созданные ими.

Для выполнения данной задачи в основные таблицы, к которым производится доступ пользователя включены дополнительные поля, хранящие имена пользователей, которые создают записи.

Политики контроля строк добавлены и активированы во все основные таблицы, кроме таблицы «Парикмахерские», поскольку доступ к ней имеют только пользователи группы директоров, каждый из которых имеет полное право на управление всеми записями данной таблицы.

## 4.6 Партицирование одной из основных таблиц БД

При разработке БД проведено партицирование таблицы «Журнал» по полю «Пользователь», отражающему того пользователя, который добавлял запись в таблицу, которое также используется и для обеспечения безопасности на уровне строк. Подобное разделение позволит ускорить взаимодействие пользователей-работников с созданными ими данными (см. рис. 4.6).

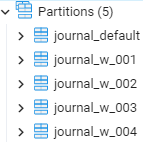


Рисунок 4.6 – Партиции таблицы «Журнал»

Создание партиции происходит сразу после создания новой роли работника, при удалении пользователя все данные из соответствующей ему партиции перемещаются в стандартную партицию.

## 4.7 Проектирование запросов к базе данных

Исходя из поставленной задачи, было создано 20 запросов, из который 10 – простые запросы, и ещё 10 – сложные или итоговые запросы. Образцы созданных запросов представлены далее:

- Четыре запроса, использующих внутреннее соединение с условиями по внешнему ключу или по дате (см. пример на рис. 4.7 - 4.8).

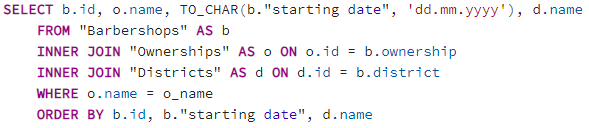


Рисунок 4.7 – Код одного из запросов, использующих внутреннее

соединение с условиями

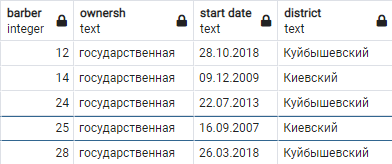


Рисунок 4.8 – Результат выполнения одного из запросов,

использующих внутреннее соединение с условиями

- Три запроса, использующих внутреннее соединение без условия (см. пример на рис. 4.9 - 4.10).

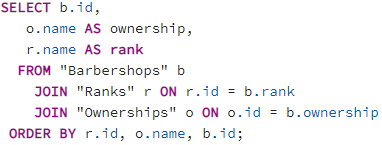


Рисунок 4.9 – Код одного из запросов, использующих внутреннее

соединение без условий

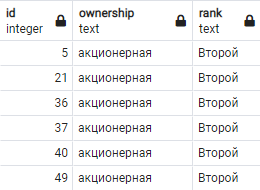


Рисунок 4.10 – Результат выполнения одного из запросов,

использующих внутреннее соединение без условий

- Запрос, использующий левое внешнее соединение (см. рис. 4.11 - 4.12).

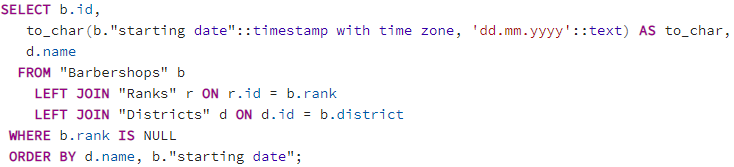


Рисунок 4.11 – Код запроса, использующего левое внешнее соединение

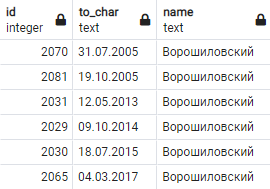


Рисунок 4.12 – Результат выполнения запроса, использующего

левое внешнее соединение

- Запрос, использующий правое внешнее соединение (см. рис. 4.13 - 4.14).

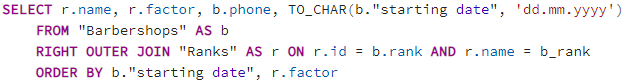


Рисунок 4.13 – Код запроса, использующего правое внешнее соединение

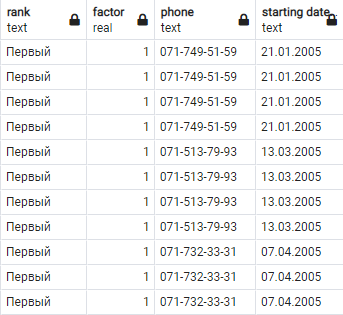


Рисунок 4.14 – Результат выполнения запроса, использующего

правое внешнее соединение

- Запрос на запросе, созданном по принципу левого внешнего соединения (см. рис. 4.15 - 4.16).

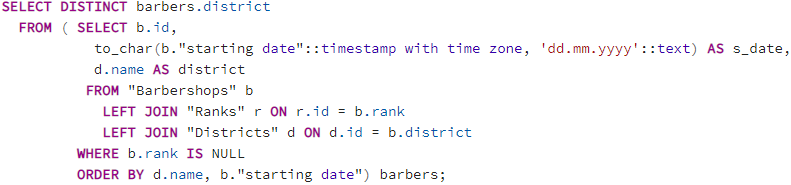


Рисунок 4.15 – Код запроса на запросе, созданном по принципу левого внешнего

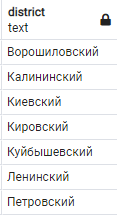


Рисунок 4.16 – Результат выполнения запроса на запросе,

созданном по принципу левого внешнего

- Шесть итоговых запросов (см. пример на рис. 4.17 - 4.18).

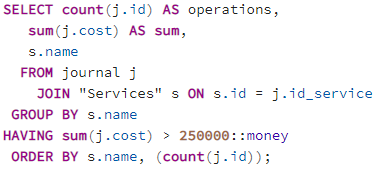


Рисунок 4.17 – Код одного из итоговых запросов

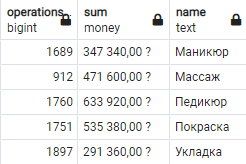


Рисунок 4.18 – Результат выполнения одного из итоговых запросов

- Запрос на запросе по принципу итогового запроса (см. рис. 4.19 - 4.20).

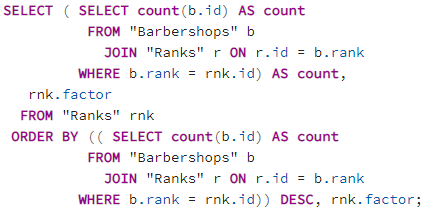


Рисунок 4.19 – Код запроса на запросе, созданном по принципу итогового

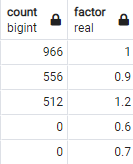


Рисунок 4.20 – Результат выполнения запроса на запросе,

созданном по принципу итогового

- Запрос с использованием объединения (см. рис. 4.21 - 4.22).

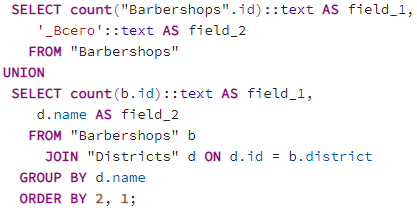


Рисунок 4.21 – Код запроса с использованием объединения

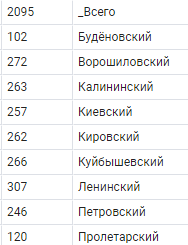


Рисунок 4.22 – Результат выполнения запроса с использованием объединения

- Два запроса с подзапросами (с использованием in, not in, case, операциями над итоговыми данными) (см. пример на рис. 4.23 - 4.24).

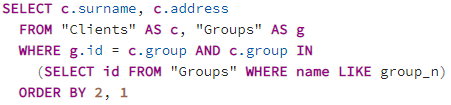


Рисунок 4.23 – Код одного из запросов с подзапросом



Рисунок 4.24 – Результат выполнения одного из запросов с подзапросом

## 4.8 Создание представлений и хранимых процедур, функций

Все созданные запросы, которые требуют ввода конкретизирующих данных от пользователя, оформлены в виде функций, другие, что собирают необходимую информацию из таблиц и сразу передают её, – в виде представлений.

Также создано модифицируемое представление, которое полностью отражает таблицу «Клиенты», исключая из выбранных строк содержащую «нулевого» клиента. Модификация таблицы клиентов через данное представление происходит при помощи триггеров вместо взаимодействия, которые применяют данные взаимодействия для вызова соответствующего взаимодействия с родительской таблицей (см. рис. 4.25 - 4.26).



Рисунок 4.25 – Код тела модифицируемого представления

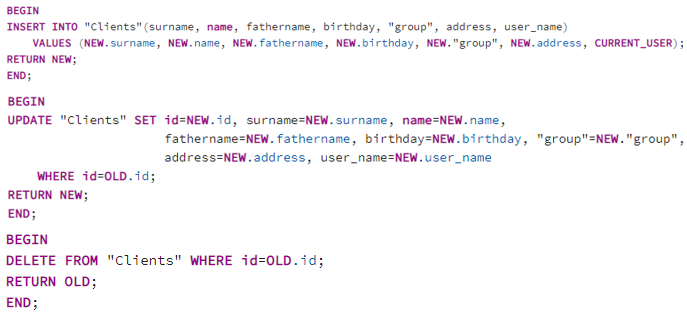


Рисунок 4.26 – Коды тел триггеров модифицируемого представления

Из дополнительных функций созданы:

- Функция, выдающая родительские роли указанного пользователя, используемая для определения группы, к которой относится авторизуемый пользователь (см. рис. 4.27).

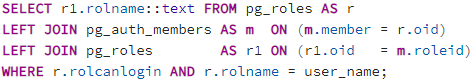


Рисунок 4.27 – Код тела функции родительских ролей пользователя

# 5 РАЗРАБОТКА КЛИЕНТСКОГО ПРИЛОЖЕНИЯ

## 5.1 Формы и компоненты для работы в роли «Менеджер»

## 5.2 Формы и компоненты для работы в роли «Директор»

## 5.3 Формы и компоненты для работы в роли «Работник»

## 5.4 Генерация результатов не менее трех итоговых запросов (диаграммы, экспорт в Excell)

# 6 ТЕСТИРОВАНИЕ РАЗРАБОТАННОЙ ИНФОРМАЦИОННОЙ СИСТЕМЫ

# ВЫВОДЫ

В ходе выполнения курсовой работы получены практические навыки проектирования, моделирования и создания баз данных средствами СУБД PostgreSQL, а также разработки клиент-серверных систем контроля баз данных, основанных на многопользовательском доступе с использованием различных ролей.

В результате выполнения курсовой работы было проведено инфологическое, даталогическое и физическое проектирование модели базы данных по заданному варианту (информационная система «Парикмахерские»), был создан проект базы данных, созданы и заполнены соответствующим образом все необходимые таблицы (таблицы-справочники, вспомогательные таблицы и основные таблицы).

Также были разработаны серверное и клиентское приложения на языке программирования высокого уровня C#, обеспечивающие взаимодействие пользователя с базой данных. В программе реализована возможность сохранения результатов запросов в формат Excel.

Разработанная система создана в целях получения и закрепления навыков создания клиент-серверной информационной системы средствами СУБД и не подлежит к практическому применению, кроме демонстрации возможностей при моделировании и реализации схожих проектов во время образовательного процесса.

Для пользователей разработанной системы было написано соответствующие руководства использования.

# СПИСОК ЛИТЕРАТУРЫ

1. Базы данных. [Электронный ресурс]. URL: http://flash-library.narod.ru/Ch-Informatics/lektion/lektion7.html.

2. Системы управления базами данных (СУБД). [Электронный ресурс]. URL: http://wiki.mvtom.ru/index.php/Системы\_управления\_базами\_данных\_(СУБД)

# ПРИЛОЖЕНИЕ А. Техническое задание

ГОУ ВПО «Донецкий национальный технический университет»

Факультет Интеллектуальных систем и программирования

Кафедра "Программная инженерия" им. Л.П. Фельдмана

Утверждаю

Зори С.А.

\_\_\_\_\_\_\_\_\_\_\_

08.02.2022 г.

**ТЕХНИЧЕСКОЕ ЗАДАНИЕ**

**на курсовую работу по дисциплине**

**«Программирование систем с серверами баз данных»**

выдано студенту группы ПИ-19 «Б» Носаченко Артёму Александровичу

**Тема:** **«Создание клиент-серверной информационной системы средствами СУБД»**

**Описание предметной области:**

14. Для учета работы парикмахерских города необходима информация о парикмахерских (номер, район города, разряд (высший, первый, второй), тип собственности (частная, государственная, акционерная,…), год начала функционирования, телефон), клиентах (ФИО, дата рождения, социальная группа (предприниматель, банковский служащий, инженер, рабочий,…), домашний адрес)) и оказанных клиентам услугах (парикмахерская , название услуги (стрижка, завивка, укладка, маникюр, массаж,…), стоимость ( зависит от разряда парикмахерской), дата оказания услуги).

Донецк – 2022

**Задание на курсовую работу**

1. Спроектировать концептуальную модель базы данных (БД) для заданной предметной области и представить ее в виде взаимосвязанных таблиц, находящихся в третьей нормальной форме (в случае денормализации БД – обосновать необходимость). Выделить базовые таблицы и таблицы-справочники, указать для них первичные и внешние ключи.
2. Создать базу данных в среде СУБД средствами языка SQL. Добавить таблицы, домены, индексы.
3. Разработать не менее шести триггеров (по одному для каждого типа события), как минимум для двух различных таблиц БД. Триггеры типа BEFORE INSERT должны быть созданы для всех таблиц и с использованием генераторов задавать значение первичного ключа для вновь добавляемой записи.
4. Заполнить таблицы БД с использованием соответствующих запросов на языке SQL (не менее десяти записей в каждом справочнике, не менее 10 000 - 50 000 псевдослучайных записей в таблицах).
5. Сформулировать следующие виды запросов:

* симметричное внутреннее соединение с условием (два запроса с условием отбора по внешнему ключу, два – по датам);
* симметричное внутреннее соединение без условия (три запроса);
* левое внешнее соединение;
* правое внешнее соединение;
* запрос на запросе по принципу левого соединения;
* итоговый запрос без условия;
* итоговый запрос без условия c итоговыми данными вида: «всего», «в том числе»;
* итоговые запросы с условием на данные (по значению, по маске, с использованием индекса, без использования индекса);
* итоговый запрос с условием на группы;
* итоговый запрос с условием на данные и на группы;
* запрос на запросе по принципу итогового запроса;
* запрос с использованием объединения
* запросы с подзапросами (с использованием in, not in, case, операциями над итоговыми данными).

1. Запросы без параметров реализовать в виде представлений, остальные запросы – в виде хранимых процедур и/или функций. Создать, по меньшей мере, одно модифицируемое представление, используя механизм триггеров. ВСЯ логика проектируемого ПО – на сервере.
2. Разработать клиентское приложение, которое предоставляет следующие возможности для работы с созданной базой данных:

* многопользовательский режим работы (одна программа для всех ролей – ситуативный доступ к интерфейсу)
* наличие нескольких ролей пользователя (менеджер – добавление/удаление/редактирование пользователей, их прав/ролей; директор – просмотр отчётов о прибыли и убытках, работник – создание записей о проводимых работах, изучение личного дохода)
* просмотр содержимого таблиц и представлений (здесь и далее – с учетом прав пользователей);
* добавление, редактирование и удаление записей таблиц и модифицируемых представлений;
* работа с наборами данных, находящимися в отношении «один-ко-многим» (создать составную форму для просмотра и редактирования данных родительской и дочерней таблиц);
* поиск и фильтрация данных отображаемых таблиц;
* просмотр результатов выполнения запросов;
* визуализация результатов одного из итоговых запросов (диаграммы, экспорт в Excel).

1. Обеспечить защиту данных, информации от несанкционированного доступа, сделать защиту на уровне строк, выполнить партицирование одной из основных таблиц

**Рекомендуемое содержание пояснительной записки**

Титульный лист

Реферат

Содержание

Введение

1. Описание предметной области, постановка задачи

2. Обоснование выбора СУБД, описание возможностей СУБД

3. Обоснование выбора инструментальные средств для написания клиентской части, проектирование структуры ПО

3.1 Невизуальные компоненты для работы с данными

3.2 Визуальные компоненты отображения данных

3.3 Разработка шаблонов приложений для работы c таблицами базы данных

4. Проектирование базы данных в выбранной СУБД

4.1 Проектирование концептуальной модели БД

4.2 Создание таблиц, доменов, индексов, сиквенсов

4.3 Разработка триггеров

4.4 Организация многоролевого доступа к данным

4.5 Разграничение доступа к данным на уровне строк (в зависимости от роли и логина)

4.6 Партицирование одной из основных таблиц БД

4.7 Проектирование запросов к базе данных

4.8 Создание представлений и хранимых процедур, функций

5. Разработка клиентского приложения

5.1 Формы и компоненты для работы в роли «Менеджер»

5.2 Формы и компоненты для работы в роли «Директор»

5.3 Формы и компоненты для работы в роли «Работник»

5.4 Генерация результатов не менее трех итоговых запросов (диаграммы, экспорт в Excell)

6 Тестирование разработанной информационной системы (в т.ч. включая защиту от несанкционированного доступа, одновременную работы с данными, каскадное удаление)

Заключение/выводы и предложения

Список литературы

Приложение А. Техническое задание

Приложение Б. Листинг шаблонов

Приложение В. Листинг серверного приложения

Приложение Г. Листинг клиентского приложения

Приложение Д. Руководство пользователя

Приложение Е. Руководство суперпользователя

Приложение Ж. Руководство администратора

**График выполнения курсовой работы**

|  |  |
| --- | --- |
| Неделя | Работа |
| 1-2 | Выдача и изучение задания |
| 3 | Анализ требований к системе и способов их реализации |
| 4-5 | Проектирование и реализация БД (таблицы, домены, индексы, роли, RLS, партицирование) |
| 6-7 | Создание триггеров и заполнение таблиц БД |
| 8-9 | Создание представлений и хранимых процедур, запросов |
| 10-13 | Разработка клиентского приложения |
| 14 | Тестирование и отладка системы |
| 15 | Оформление пояснительной записки |
| 16-17 | Защита курсовой работы |

Дата выдачи задания 08.02.2022 г.

Задание принял \_\_\_\_\_\_\_\_\_\_\_\_ Носаченко А. А.

Руководители проекта \_\_\_\_\_\_\_\_\_\_\_\_ Щедрин С. В.

**\_\_\_\_\_\_\_\_\_\_\_\_** Ногтев Е. А.

# ПРИЛОЖЕНИЕ Б. Листинг серверного приложения

# ПРИЛОЖЕНИЕ В. Листинг клиентского приложения

# ПРИЛОЖЕНИЕ Г. Руководство пользователя

# ПРИЛОЖЕНИЕ Д. Руководство администратора