Министерство образования и науки

Российской федерации

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ

ОБРАЗОВАНИЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«Курганский государственный университет»

Кафедра «Программное обеспечение автоматизированных систем»

**Проект БД для магазина по продаже вычислительной техники**

**РУКОВОДСТВО АДМИНИСТРАТОРА**

**РФ** **41287462 260091 92**

**Листов 12**

Курган 2025

**АННОТАЦИЯ**

Документ предназначен для администратора базы данных и системного администратора информационной системы розничного магазина. В нём приведена информация о логической структуре базы данных, описаны роли пользователей и правила разграничения прав доступа. Также представлена пошаговая инструкция по установке базы данных.

СОДЕРЖАНИЕ

[1 НАЗНАЧЕНИЕ СИСТЕМЫ 4](#_Toc198672594)

[2 ОБЯЗАННОСТИ И ЗАДАЧИ АДМИНИСТРАТОРА 5](#_Toc198672595)

[3 ТРЕБОВАНИЯ К ОБЕСПЕЧЕНИЮ РАБОЧЕГО МЕСТА АДМИНИСТРАТОРА 6](#_Toc198672596)

[4 РОЛИ ПОЛЬЗОВАТЕЛЕЙ И ПРАВИЛА РАЗГРАНИЧЕНИЯ ДОСТУПА 7](#_Toc198672597)

[4.1 Основные роли пользователей 7](#_Toc198672598)

[4.2 Создание ролей и назначение прав 7](#_Toc198672599)

[УСТАНОВКА БАЗЫ ДАННЫХ 11](#_Toc198672600)

[ПРИЛОЖЕНИЕ А 12](#_Toc198672601)

[Схема базы данных 12](#_Toc198672602)

# 1 НАЗНАЧЕНИЕ СИСТЕМЫ

Разрабатываемая система предназначена для автоматизации процессов управление ассортиментом товаров, обработка клиентских заказов и взаимодействие с поставщиками. Система обеспечивает централизованное хранение данных, а также позволяет формировать отчёты и управлять заказами.

Категории пользователей системы:

* Клиенты (оформление и управление заказами);
* Менеджер (управление складскими запасами, прайс-листом, классификатором товаров);
* Аналитик (формирование отчетов, анализ продаж);
* Администратор системы (обслуживание базы данных, доступы и мониторинг).

# 2 ОБЯЗАННОСТИ И ЗАДАЧИ АДМИНИСТРАТОРА

Администратор базы данных отвечает за:

* создание и обслуживание базы данных;
* управление пользователями и их правами доступа;
* контроль над целостностью и актуальностью информации;
* обеспечение безопасности базы;
* резервное копирование и восстановление;
* импорт и экспорт данных;
* выполнение SQL-скриптов, обновление бизнес-логики;
* аудит выполненных операций и журналирование.

# 3 ТРЕБОВАНИЯ К ОБЕСПЕЧЕНИЮ РАБОЧЕГО МЕСТА АДМИНИСТРАТОРА

**Программное обеспечение сервера:**

1. Операционная система: Microsoft Windows 10 / Server 2019 или Linux Debian 11 / Ubuntu 20.04 LTS и выше
2. СУБД: PostgreSQL 17.5

**Программное обеспечение рабочего места:**

1. Операционная система: Windows 10 или Linux Debian 11 / Ubuntu 20.04 LTS и выше
2. DataGrip 2024.1 или выше
3. Браузер (для веб-клиентов, при необходимости): Microsoft Edge, Chrome
4. Доступ к локальной или удалённой базе данных computer\_store\_db

**4 РОЛИ ПОЛЬЗОВАТЕЛЕЙ И ПРАВИЛА РАЗГРАНИЧЕНИЯ ДОСТУПА**

В информационной системе розничного магазина реализовано разграничение доступа на уровне базы данных с использованием ролей. Это позволяет обеспечить безопасность, целостность данных и предотвратить несанкционированные действия пользователей.

**4.1 Основные роли пользователей**

В таблице 1 представлены основные роли пользователей и описаны их права в системе.

|  |  |
| --- | --- |
| **Роль** | **Назначение и права доступа** |
| client | Оформление и просмотр собственных заказов, также возможность их возврата |
| analytic | Доступ к аналитическим функциям базы для формирования отчетов статистики |
| manager | Доступ ко всем данным для управления прайс-листом, классификатором товаров и складскими запасами |
| admin | Полный доступ ко всей базе данных и объектам |

**4.2 Создание ролей и назначение прав**

Ниже приведены примеры SQL-команд, которые следует выполнить в DataGrip для создания ролей и предоставления прав доступа.

**Создание ролей:**  
CREATE ROLE client;

CREATE ROLE analytic;

CREATE ROLE manager;

CREATE ROLE admin;

**Назначение прав конкретным ролям:**

-- Права для клиента

GRANT SELECT ON analitics.t\_discount TO client;

GRANT SELECT ON clients.t\_client TO client;

GRANT SELECT ON products.t\_model TO client;

GRANT SELECT ON products.t\_manufacturer TO client;

GRANT SELECT ON categories.t\_category\_property TO client;

GRANT SELECT ON products.t\_product\_property TO client;

GRANT SELECT ON products.t\_product TO client;

GRANT SELECT ON categories.t\_category TO client;

GRANT SELECT ON storages.t\_inventory TO client;

GRANT SELECT ON products.t\_product\_instance TO client;

GRANT SELECT, INSERT, UPDATE, DELETE ON clients.t\_basket\_info TO client;

GRANT EXECUTE ON FUNCTION products\_get\_by\_category (integer) TO client;

GRANT EXECUTE ON FUNCTION inventory\_get\_product\_count (integer) TO client;

GRANT EXECUTE ON FUNCTION basket\_info\_set (integer, integer, integer) TO client;

GRANT EXECUTE ON FUNCTION get\_product\_discount(text[], integer) TO client;

GRANT EXECUTE ON FUNCTION get\_category\_discount (integer) TO client;

GRANT EXECUTE ON FUNCTION client\_get\_discount (

integer,

numeric,

numeric,

double precision

) TO client;

-- Права для аналитика

GRANT SELECT ON products.t\_model TO analytic;

GRANT SELECT ON products.t\_manufacturer TO analytic;

GRANT SELECT ON categories.t\_category\_property TO analytic;

GRANT SELECT ON products.t\_product\_property TO analytic;

GRANT SELECT ON products.t\_product TO analytic;

GRANT SELECT ON categories.t\_category TO analytic;

GRANT SELECT ON storages.t\_inventory TO analytic;

GRANT SELECT ON products.t\_product\_instance TO analytic;

GRANT EXECUTE ON FUNCTION get\_sales\_analytics (

timestamp

with

time zone,

timestamp

with

time zone,

text

) TO analytic;

-- Права для менеджера

GRANT SELECT, INSERT, UPDATE ON products.t\_model TO manager;

GRANT SELECT, INSERT , UPDATE ON products.t\_manufacturer TO manager;

GRANT SELECT, INSERT , UPDATE ON categories.t\_category\_property TO manager;

GRANT SELECT, INSERT , UPDATE ON products.t\_product\_property TO manager;

GRANT SELECT, INSERT , UPDATE ON products.t\_product TO manager;

GRANT SELECT, INSERT , UPDATE ON categories.t\_category TO manager;

GRANT SELECT, INSERT , UPDATE ON storages.t\_inventory TO manager;

GRANT SELECT ON products.t\_product\_instance TO manager;

GRANT SELECT, INSERT, UPDATE ON analitics.t\_discount TO manager;

GRANT EXECUTE ON FUNCTION products\_get\_by\_category (integer) TO manager;

GRANT EXECUTE ON FUNCTION inventory\_get\_product\_count (integer) TO manager;

GRANT EXECUTE ON FUNCTION get\_product\_discount(text[], integer) TO manager;

GRANT EXECUTE ON FUNCTION get\_category\_discount (integer) TO manager;

-- Права для администратора

GRANT ALL PRIVILEGES ON DATABASE computer\_store\_db TO admin;

**Добавление пользователя в роль**

-- Добавление логина/пользователя к роли (указать имя логина и базы данных)

CREATE ROLE client LOGIN PASSWORD ‘stu\_Client\_214sfe’;

CREATE ROLE manager LOGIN PASSWORD ‘stu\_Manager\_323gfd’;

CREATE ROLE analytic LOGIN PASSWORD ‘stu\_Analytic\_891ios’;

CREATE ROLE admin SUPERUSER LOGIN PASSWORD ‘stu\_Admin\_234sji’;

# УСТАНОВКА БАЗЫ ДАННЫХ

Процесс установки базы данных состоит из следующих шагов:

1. Установка PostgreSQL 17.5;
2. Запустить среду DataGrip и выполнить соединение с сервером базы данных;
3. Создать новую пустую базу данных, куда нужно импортировать данные;
4. Щелкнуть правой кнопкой мыши на «PostgreSQL» в окне «DatabaseExplorer», «Create», «Database» дать имя базе данных и нажать кнопку «ОК».
5. Импортировать данные из файла «computer\_store\_db-dump.sql» в новую базу данных: щелкните правой кнопкой мыши на базу данных, выберите «Run SQL Script…» или «Import Data from File…», Выберите файл computer\_store\_db-dump.sql .

# ПРИЛОЖЕНИЕ А

## **Схема базы данных**

