**Федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

**«Саратовский государственный технический университет**

**имени Гагарина Ю.А.»**

Институт прикладных информационных технологий и коммуникаций

Кафедра Прикладные информационные технологии

**Курсовая работа**

по дисциплине «Технологии управления данными в объектно-реляционных СУБД»

на тему «Сеть продуктовых магазинов»

Выполнил студент группы Б1-ИФСТ-32

Чермашенцев Андрей Павлович

Проверил: доцент кафедры ПИТ, к.ф.-м.н.

Леванде Алексей Борисович

Комиссия по защите:

Доцент кафедры ПИТ, к.ф.-м.н. Леванде А.Б.

Курсовая работа защищена на оценку «\_\_\_\_\_\_\_\_\_\_\_\_\_\_»

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Леванде А.Б.

(дата, подпись члена комиссии)

Саратов 2024

**Замечания**

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

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Леванде А.Б.

(дата, подпись члена комиссии)

**Федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

**«Саратовский государственный технический университет**

**имени Гагарина Ю.А.»**

**Кафедра «Прикладные информационные технологии»**

ЗАДАНИЕ

на выполнение курсовой работы

по дисциплине «Интеллектуальные информационные системы и технологии»

студенту ИнПИТ группы Б1-ИФСТ-32

В курсовой работе необходимо:

* Построить логическую модель данных
* Описать логическую модель
* Оценить качества модели данных на основе анализа зависимостей
* Реализовать приложение для работы с моделью данных

Дата выдачи: «01» октября 2024 г.

Срок выполнения: «20» декабря 2024 г.

Руководитель: \_\_\_\_\_\_\_\_\_\_\_\_\_ доцент кафедры ПИТ, к.ф.-м.н. Леванде А.Б.

Студент: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Чермашенцев А. П.

**Содержание**

[**Постановка задачи и описание предметной области** 5](#_Toc185396898)

[**Логическая модель данных (ER-диаграмма)** 6](#_Toc185396899)

[**Описание логической модели данных** 8](#_Toc185396900)

[**Оценка качества модели данных на основе анализа функциональных зависимостей** 10](#_Toc185396901)

[**Анализ связей между отношениями** 11](#_Toc185396902)

# **Постановка задачи и описание предметной области**

**Постановка задачи:**

Рассматриваемая предметная область связана с управлением товарными запасами и движением товаров между складами и магазинами. Основная цель – обеспечить эффективный учет и контроль поступления, хранения, выдачи и продажи товаров. Необходимо разработать базу данных для хранения информации о товарах, поставках, продажах, а также запросах магазинов на дополнительные поставки. База данных должна позволять:

* Хранить сведения о магазинах (адрес, контактные данные, управляющий).
* Фиксировать поступление товаров (накладные на поставку, детализирующие позиции товара).
* Учитывать продажи (какой товар, в каком количестве и в какой день был продан).
* Отслеживать запросы от магазинов на необходимый товар (запросы на поставку).
* Отображать состояние складских запасов и движение товаров по различным точкам (складам и магазинам).

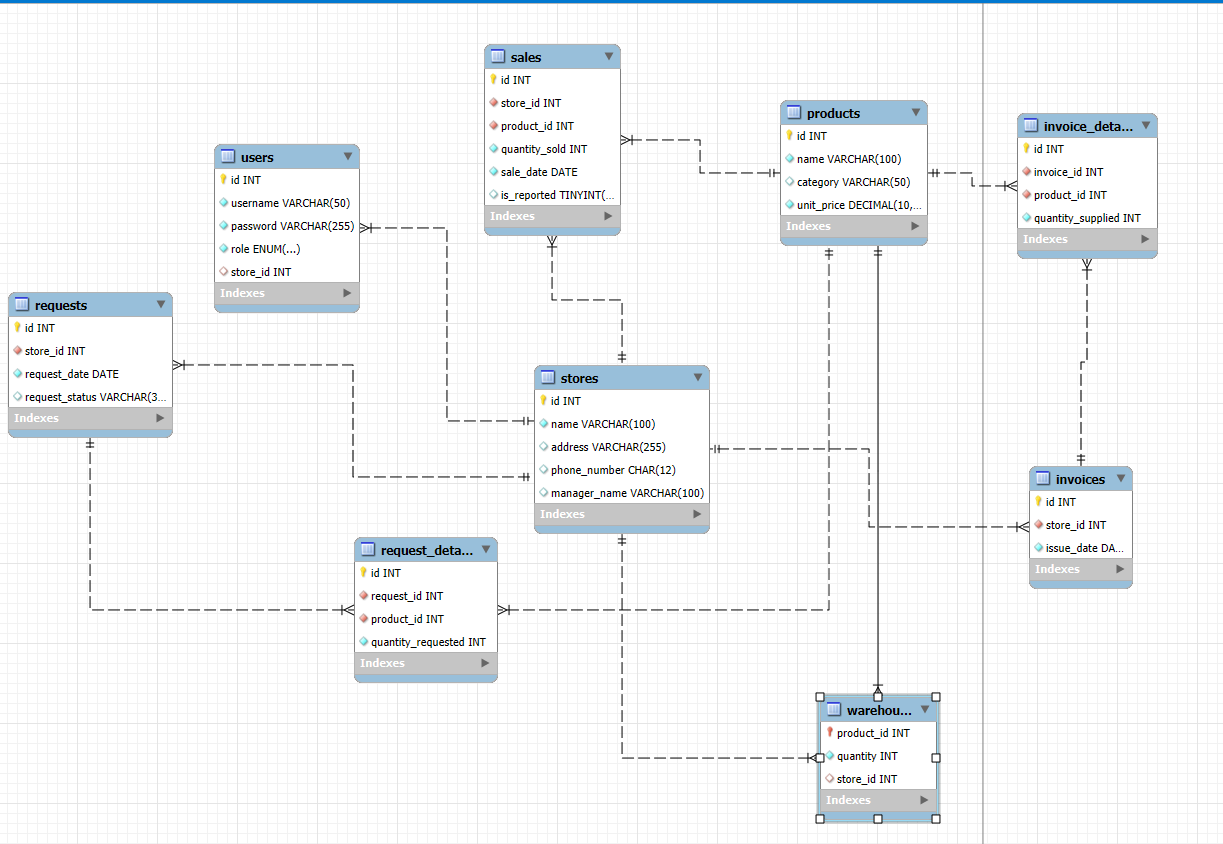
**Описание предметной области:**

Сеть магазинов реализует различные товары. Каждый магазин периодически формирует запросы на поставку определённого товара в определённом количестве. Эти запросы регистрируются в системе и после одобрения формируются накладные (invoices) на поставку. Товары имеют свои характеристики (название, категория, цена), а каждый магазин имеет свой адрес, контактные данные и управляющего.

Система должна предоставлять возможность контролировать склады, формировать отчёты по продажам, отслеживать актуальные остатки товара и анализировать запросы магазинов.

# **Логическая модель данных (ER-диаграмма)**

Ниже приводится изображение (рис.1) и описание сущностей и связей между ними.



*Рисунок 1 – ER-диаграмма приложения*

Данная ER-диаграмма отражена на предоставленном изображении.

**Основные сущности и связи:**

* users (пользователи): хранит данные о пользователях системы (например, менеджерах или операторах), их ролях и какой магазин они представляют.
* stores (магазины): содержит данные о магазинах сети (название, адрес, контактный номер, управляющий).
* products (товары): справочник товаров (наименование, категория, цена).
* sales (продажи): фиксирует факты продажи товаров в конкретном магазине в определенный день.
* invoices (накладные): отражает поставки товаров в магазины, с указанием даты поставки.
* invoice\_details (детали накладных): детализация позиций товара в каждой накладной с указанием количества.
* requests (запросы): отражает запросы магазинов на поставку определённых товаров.
* request\_details (детали запросов): детализация товаров и количества в конкретном запросе.
* warehouse (склад): хранит информацию о количестве товаров, доступных на складе для каждого магазина.

**Схема связей:**

* Один магазин может иметь много продаж (1:М связь между stores и sales).
* Один магазин может получать множество накладных поставок (1:М между stores и invoices).
* Каждая накладная может содержать много позиций товаров (1:М между invoices и invoice\_details).
* Каждый товар может фигурировать во многих продажах, накладных и запросах (1:М между products и sales, products и invoice\_details, products и request\_details).
* Один магазин может инициировать множество запросов (1:М между stores и requests).
* Каждый запрос может содержать множество позиций товаров (1:М между requests и request\_details).
* Таблица warehouse связывает продукт и магазин, отражая текущее количество на складе (1:М, один магазин – множество записей о разных товарах).

# **Описание логической модели данных**

Данные в модели распределены по таблицам, соответствующим сущностям предметной области. Ниже дано краткое описание основных полей:

1. **users**:

* id (PK) – уникальный идентификатор пользователя.
* username – логин пользователя.
* password – пароль (хранится в зашифрованном виде).
* role – роль пользователя (например, ‘admin’, ‘manager’).
* store\_id (FK) – ссылка на магазин, с которым связан пользователь.

1. **stores**:

* id (PK) – уникальный идентификатор магазина.
* name – название магазина.
* address – адрес магазина.
* phone\_number – контактный номер.
* manager\_name – имя управляющего магазином.

1. **products**:

* id (PK) – уникальный идентификатор товара.
* name – наименование товара.
* category – категория товара (например, ‘electronics’, ‘furniture’).
* unit\_price – стоимость за единицу товара.

1. **sales**:

* id (PK) – уникальный идентификатор записи о продаже.
* store\_id (FK) – ссылка на магазин, где произошла продажа.
* product\_id (FK) – ссылка на товар.
* quantity\_sold – количество проданного товара.
* sale\_date – дата продажи.
* is\_reported – флаг, отмечающий, отражена ли продажа в отчётах.

1. **invoices**:

* id (PK) – уникальный идентификатор накладной.
* store\_id (FK) – ссылка на магазин, который получает товары.
* issue\_date – дата формирования накладной.

1. **invoice\_details**:

* id (PK) – уникальный идентификатор строки в накладной.
* invoice\_id (FK) – ссылка на накладную, к которой относится позиция.
* product\_id (FK) – ссылка на товар.
* quantity\_supplied – количество поставленного товара по этой позиции.

1. **requests**:

* id (PK) – уникальный идентификатор запроса.
* store\_id (FK) – ссылка на магазин, инициировавший запрос.
* request\_date – дата формирования запроса.
* request\_status – статус запроса (например, ‘pending’, ‘approved’, ‘denied’).

1. **request\_details**:

* id (PK) – уникальный идентификатор строки запроса.
* request\_id (FK) – ссылка на запрос, к которому относится строка.
* product\_id (FK) – ссылка на товар.
* quantity\_requested – количество необходимого товара.

1. **warehouse**:

* product\_id (FK) – ссылка на товар.
* store\_id (FK) – ссылка на магазин или склад.
* quantity – текущее количество товара на складе.

# **Оценка качества модели данных на основе анализа функциональных зависимостей**

Анализ отношений Пользователи (users):

username – текстовый логин, под которым пользователь входит в систему, нужен для аутентификации; password – строковое поле для хранения хэшированного пароля, обеспечивает безопасность доступа; role – атрибут типа перечисления, указывающий роль пользователя (например, администратор, менеджер), что важно для разграничения прав доступа; store\_id – числовая ссылка на конкретный магазин, к которому пользователь относится, упрощает фильтрацию данных по магазину.

Все неключевые атрибуты (логин, пароль, роль, идентификатор магазина) функционально зависят только от id пользователя и не зависят друг от друга. Таким образом, в таблице отсутствуют частичные и транзитивные зависимости.

Таблица users удовлетворяет по меньшей мере третьей нормальной форме. Декомпозиция не требуется.

Анализ отношений Магазины (stores):

name – строковое название магазина, нужно для отображения и поиска; address – текстовый адрес, чтобы понимать физическое расположение торговой точки; phone\_number – контактный номер телефона, чтобы связаться с персоналом; manager\_name – имя управляющего, чтобы знать, кто отвечает за работу магазина.

Все перечисленные неключевые атрибуты зависят напрямую от id магазина. Транзитивных зависимостей нет, так как ни один неключевой атрибут не определяет другие неключевые атрибуты. Отношение stores соответствует как минимум 3НФ. Избыточность минимальна, дальнейшая декомпозиция не требуется.

Анализ отношений Продукты (products):

name – наименование товара, позволяющее его идентифицировать и отображать для пользователя; category – название категории, к которой относится товар, упрощает группировку и поиск; unit\_price – цена за единицу, необходима для расчёта итоговой стоимости при продажах и поставках.

Все данные о товаре (название, категория, цена) функционально зависят только от id товара. Нет зависимостей между неключевыми атрибутами, следовательно, отсутствуют транзитивные зависимости. Таблица products удовлетворяет требованиям 3НФ и не требует декомпозиции.

Анализ отношений Продажи (sales):

store\_id – ссылка на магазин, где произошла продажа, для учёта продаж по точкам; product\_id – ссылка на конкретный товар, чтобы понимать, что именно продано; quantity\_sold – число проданных единиц товара, нужно для обновления остатков; sale\_date – дата продажи, важна для анализа динамики продаж; is\_reported – логический флаг, указывающий на то, включена ли продажа в отчётность.

Все атрибуты, описывающие сделанную продажу (идентификатор магазина, идентификатор товара, количество, дата продажи и флаг отчетности), зависят от id записи о продаже. Транзитивных или частичных зависимостей нет, поскольку ключ простой и неразложимый. Отношение sales удовлетворяет по меньшей мере 3НФ. Дальнейшая декомпозиция не требуется.

Анализ отношений Накладные (invoices):

store\_id – ссылка на магазин-получатель, чтобы знать, куда была произведена поставка; issue\_date – дата оформления накладной, важна для отслеживания истории поставок.

Атрибуты store\_id (куда поставляется товар) и issue\_date (дата оформления накладной) зависят напрямую от id накладной. Других зависимостей нет. Отношение invoices соответствует как минимум 3НФ. Декомпозиция не требуется.

Анализ отношений Детали накладных (invoice\_details):

invoice\_id – ссылка на конкретную накладную, чтобы понять, к какому документу относится данная позиция; product\_id – товар, включённый в накладную, определяет, что поставлено; quantity\_supplied – количество поставленного товара, важно для обновления остатков и расчётов.

Атрибут quantity\_supplied и связка invoice\_id-product\_id зависят от id записи в деталях накладной. Нет частичных или транзитивных зависимостей. Отношение invoice\_details соответствует 3НФ. Дальнейшая декомпозиция не требуется.

Анализ отношений Заявки (requests):

store\_id – ссылка на магазин, инициировавший заявку, чтобы знать источник потребности; request\_date – дата оформления заявки, нужна для планирования поставок; request\_status – строковый статус (например, «pending»), чтобы понимать, на какой стадии рассмотрения находится заявка.

Все данные о заявке зависят от id запроса. Нет зависимостей между неключевыми атрибутами. Отношение requests соответствует 3НФ. Декомпозиция не требуется.

Анализ отношений Деталей заявки (request\_details):

request\_id – ссылка на конкретную заявку, определяет принадлежность позиции к запросу; product\_id – товар, который требуется, позволяет определить, что именно запрашивается; quantity\_requested – количество требуемого товара, помогает планировать поставки.

Все атрибуты, связанные с конкретной деталью заявки (продукт и количество), зависят от id этой детали. Частичных или транзитивных зависимостей нет. Отношение request\_details соответствует как минимум 3НФ. Декомпозиция не требуется.

Анализ отношений Склад (warehouse):

product\_id – ссылка на товар, чтобы понимать, о каких остатках идёт речь; store\_id – ссылка на конкретный магазин или склад, определяет, где именно хранится данный товар; quantity – текущее количество единиц данного товара в запасе, важно для корректного управления запасами и планирования поставок.

Атрибут quantity (количество товара на складе) полностью зависит от всей пары (product\_id, store\_id), а не от отдельного атрибута. Таким образом, нет частичных зависимостей. Отношение warehouse соответствует 3НФ. Декомпозиция не требуется.

Каждое рассмотренное отношение в модели данных обладает ключом, от которого напрямую зависят все неключевые атрибуты. Не выявлено частичных или транзитивных зависимостей. Следовательно, вся модель данных соответствует по меньшей мере третьей нормальной форме. Это гарантирует минимальную избыточность и отсутствие аномалий при работе с данными, исключая необходимость дальнейшей декомпозиции.

# **Анализ связей между отношениями**

Анализ связей между таблицами показывает следующие типы отношений:

* Между stores и users (1:М): Один магазин может иметь нескольких пользователей (менеджеров или кассиров), но каждый пользователь относится только к одному магазину.
* Между stores и sales (1:М): Один магазин может иметь множество записей о продажах, но каждая продажа относится к конкретному магазину.
* Между products и sales (1:М): Один товар может продаваться неоднократно, каждая продажа связана с конкретным товаром.
* Между stores и invoices (1:М): Один магазин может получить много накладных, но каждая накладная привязана к одному магазину.
* Между invoices и invoice\_details (1:М): Одна накладная может содержать несколько позиций товаров, но каждая деталь относится к одной конкретной накладной.
* Между products и invoice\_details (1:М): Один товар может фигурировать в разных накладных (и их деталях), но каждая деталь относится к одному товару.
* Между stores и requests (1:М): Магазин может создавать множество запросов, но каждый запрос привязан к одному магазину.
* Между requests и request\_details (1:М): Один запрос может содержать много позиций (разных товаров), но каждая деталь привязана к одному запросу.
* Между products и request\_details (1:М): Один товар может фигурировать во множестве запросов (их деталях), но каждая конкретная запись связана с одним товаром.
* Между warehouse и (products, stores) (М:М реализовано через сущность warehouse): Каждый товар может находиться в остатках разных магазинов, и каждый магазин может иметь остатки многих товаров. Таблица warehouse решает эту связь, указывая количество товара на конкретном складе/в магазине.

Сформированная модель данных имеет чёткую структуру и адекватно отражает связи между сущностями предметной области. Это позволит эффективно управлять данными, выполнять аналитические запросы и поддерживать целостность информации.