

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ  
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Московский институт электроники и математики им. А. Н. Тихонова  
Департамент компьютерной инженерии

**ДОМАШНЯЯ РАБОТА**

По курсу: Базы данных

Тема: **Проектирование реляционной базы данных секонд-хенд магазина**

**Студенты:** Пыжов Илья Игоревич,  
Татарина Полина Юрьевна

**Группа:** БИВ225

**Дата сдачи:** 15.05.2024

**Преподаватель:** Карпова Ирина Петровна

**МОСКВА 2024**

## Оглавление

Инфологическое проектирование .....	3
Анализ предметной области .....	3
Анализ информационных задач и круга пользователей системы .....	6
Логическое проектирование реляционной БД .....	6
Преобразование ER–диаграммы в схему базы данных .....	6
Составление реляционных отношений .....	7
Нормализация полученных отношений (до 4НФ) .....	10
Определение дополнительных ограничений целостности .....	16
Описание групп пользователей и прав доступа .....	16
Реализация проекта базы данных .....	17
Создание таблиц .....	17
Создание представлений (готовых запросов) .....	22
Назначение прав доступа .....	24
Создание триггеров .....	25
Создание индексов .....	26
Разработка стратегии резервного копирования .....	26

# Инфологическое проектирование

## Анализ предметной области

База данных создается для информационного обслуживания сотрудников секонд-хенд магазина. В создаваемой БД должна храниться вся необходимая информация для непрерывной работы магазина и успешного его функционирования: информация о сотрудниках, поставках, поставщиках, покупках и покупателях магазина.

Учитывая особенности выбранной предметной области, выделим сущности и укажем их атрибуты:

### 1) Сотрудники:

- ФИО
- Паспортные данные (серия и номер)
- Дата рождения
- Пол
- ИНН
- СНИЛС
- Адрес проживания
- Телефон
- Эл. почта
- Образование
- Должность
- Оклад

В зависимости от должности назначается оклад. Должности: администратор, консультант-кассир, уборщик, грузчик.

Консультанты-кассиры обслуживают и регистрируют покупателей (выдают им их номер) и оформляют продажу товаров. Администратор заключает договоры с поставщиками.

### 2) Поставщики:

- Название (ФИО)
- Тип (фирма/частное лицо)
- ИНН поставщика
- Адрес
- ФИО руководителя
- Телефон
- Эл. почта
- Условия поставки
- Дата начала сотрудничества
- Дата окончания сотрудничества

Поставлять могут фирмы и частные лица, в случае если поставщиком является частное лицо в названии указывается ФИО, а ФИО руководителя совпадает с ФИО. В условиях поставки может указываться: «ежемесячная», «единоразовая» и др. Дата окончания сотрудничества не указывается, если сотрудничество не окончено.

### 3) Товары:

- Артикул
- Категория
- Наличие
- Цена
- Размер
- Пол
- Бренд
- Превалирующий материал
- Цвет
- Изношенность (от 1 до 5, где 1 — новая вещь, 5 — изношенная)
- Особенности

Каждый товар является уникальным в свете специфики магазина б/у вещей.

Примеры категории товаров: «головные уборы», «обувь» и др. В наличии: «0», если нет, «1», если товар есть в наличии. В качестве материала указывается превалирующий материал (большая часть от общего состава). Изношенность указывается от «1» до «5», где «1» — новая вещь, «5» — изношенная. Оценивается администратором при приеме поставки. В особенностях можно указать отличительную черту товара, например, «оторван бант на правом рукаве».

### 4) Покупатели:

- Идентифицирующий номер
- ФИО
- Телефон
- Эл. почта
- Адрес

Покупатели могут делать заказы, позвонив или написав на почту сотруднику магазина, при этом они указывают свои данные (ФИО, Телефон, Эл. почта, Адрес доставки (в случае, когда требуется доставка)) и получают идентифицирующий номер. Покупатели, совершающие покупку оффлайн, просто получают идентифицирующий номер и при необходимости и желании указывают данные, например, необходимые для доставки, либо же, чтобы иметь возможность отслеживать свои покупки.

Определим виды связей, характеризующих предметную область:

Таблица 1. Виды связей в ПрО

<i>Особенности ПрО</i>	<i>Описание связи</i>	<i>Примечание</i>
У каждого сотрудника может быть один начальник; Под руководством каждого начальника могут работать несколько сотрудников	Связь <b>начальник</b> типа один-ко-многим между разными сотрудниками	Связь необязательна с обеих сторон

Каждый сотрудник обслуживает нескольких покупателей за смену; Покупатель может быть обслужен одним сотрудником	Связь <b>обслуживать</b> типа один-ко-многим между сотрудниками и покупателями	Связь обязательная со стороны покупателя и не обязательна со стороны сотрудника (Сотрудник может не обслужить ни одного покупателя, а покупатель должен быть обслужен, чтобы совершить покупку)
Каждый сотрудник может продать несколько товаров; Товар может быть продан только одним продавцом	Связь <b>продавать</b> типа один-ко-многим между сотрудниками и товарами	Связь необязательна с обеих сторон (Сотрудник может не продать ни одного товара, товар может быть не продан)
Каждый сотрудник может заключить договор с несколькими поставщиками; Поставщик заключает договор с конкретным сотрудником	Связь <b>заключать</b> договор типа один-ко-многим между сотрудниками и поставщиками	Связь обязательная со стороны сотрудника и обязательна со стороны поставщика (Сотрудник может не заключить ни одного договора, а поставщик должен заключить договор, чтобы поставлять товар в магазин)
Каждый покупатель может купить несколько товаров; Каждый товар может быть куплен лишь одним покупателем	Связь <b>покупать</b> типа один-ко-многим между покупателями и товарами	Связь обязательна со стороны покупателя и не обязательна со стороны товара (Покупатель должен что-то купить, чтобы считаться покупателем, а товар может быть не куплен)
Каждый поставщик может поставлять несколько товаров; Каждый товар может быть поставлен лишь одним поставщиком	Связь <b>поставлять</b> типа один-ко-многим между товарами и поставщиками	Связь обязательна со стороны товаров, не обязательна со стороны поставщиков (Товар должен быть поставлен, чтобы оказаться в магазине, а поставщик на момент заключения договора может не поставлять товар в течение какого-то времени)

Исходя из выявленных связей, построим ER-диаграмму (см. рисунок 1).  
Пометки у линий на диаграмме отражают кардинальность связи: 1:N (один-ко-многим). Обязательные связи выделены более толстыми линиями.

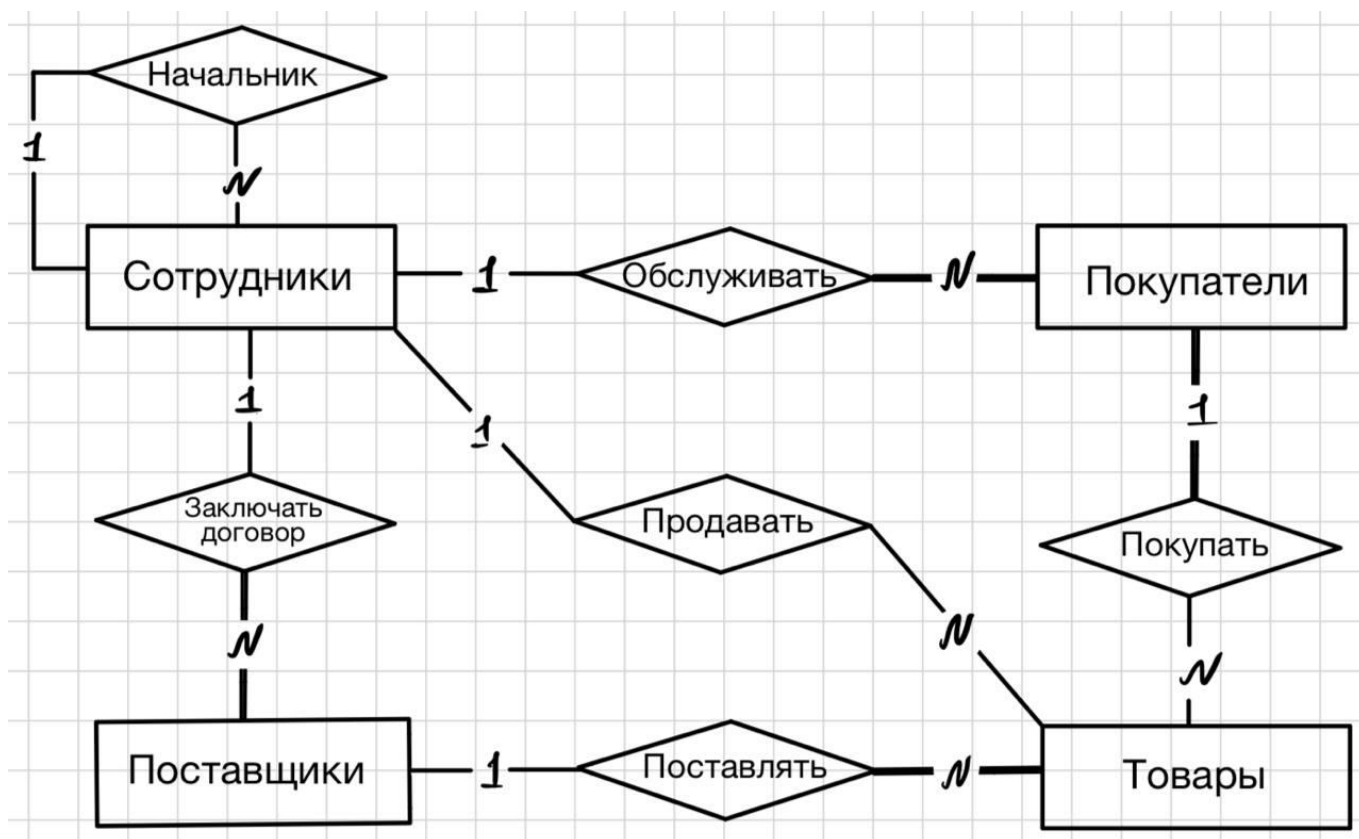


Рисунок 1. ER-диаграмма ПрО «Секонд-хенд магазин»

## Анализ информационных задач и круга пользователей системы

Определим группы пользователей, их основные задачи и запросы к БД:

### 1) Администратор:

- заключение договоров с поставщиками
- ведение учета товара
- наем новых сотрудников
- назначение оклада
- получение информации о покупателях и совершенных покупках
- получение информации о поставках и поставщиках

### 2) Консультант-кассир:

- оформление новых продаж
- регистрация новых покупателей в базе данных
- получение полной информации о товарах
- получение своей персональных данных

### 3) Уборщик, грузчик:

- получение своей персональных данных

## Логическое проектирование реляционной БД

### Преобразование ER–диаграммы в схему базы данных

База данных создается на основании схемы базы данных. Преобразование ER–диаграммы в схему БД выполняется путем сопоставления каждой сущности и каждой связи, имеющей атрибуты, отношения (таблицы) БД. Связь типа 1:N (один-ко-многим) между отношениями реализуется через внешний ключ. Ключ вводится

для дочернего отношения. Внешнему ключу должен соответствовать первичный или уникальный ключ основного (родительского) отношения.

Полученная схема реляционной базы данных (РБД):

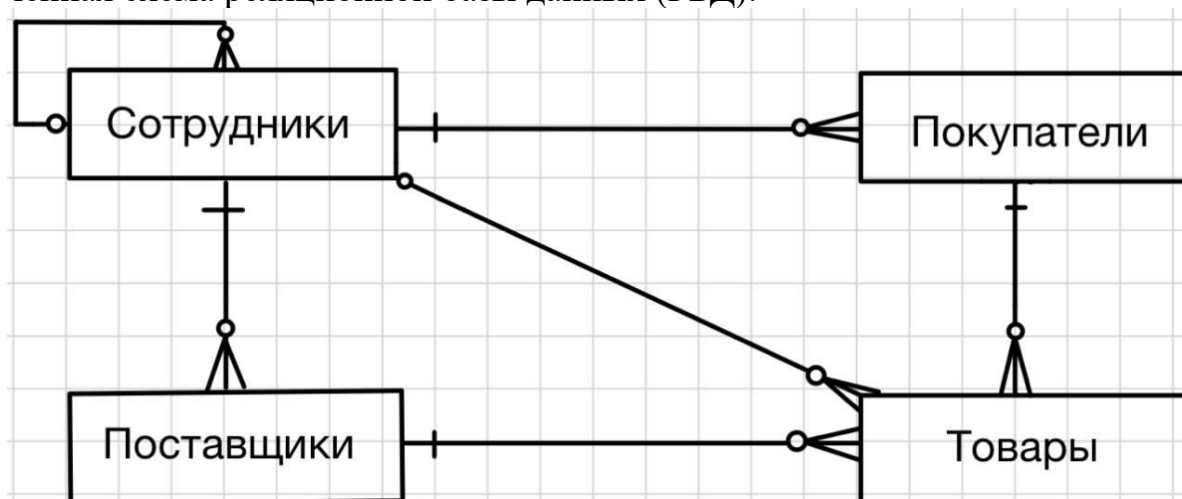


Рисунок 2. Схема РБД, полученная из ER-диаграммы

## Составление реляционных отношений

Каждое реляционное отношение соответствует одной сущности (объекту ПрО) и в него вносятся все атрибуты этой сущности. Для каждого отношения определяются первичный ключ и внешние ключи (в соответствии со схемой БД). В том случае, если отношение не имеет потенциальных ключей, вводится суррогатный (искусственный) первичный ключ.

Отношения приведены в Таблица 2-5. Для каждого отношения указаны атрибуты с их внутренним названием, типом и длиной. Типы данных обозначаются так: N – числовой, С – символьный тип фиксированной длины, V – символьный тип переменной длины, D – дата.

Отношение **Поставщики (Suppliers)**:

Таблица 2. Схема отношения **Поставщики (Suppliers)**

Содержание поля	Имя поля	Тип, длина	Примечания
Название (ФИО)	supplier_name	V(50)	обязательное поле, в случае если поставщиком является частное лицо в названии указывается ФИО
Тип	supplier_type	V(12)	обязательное поле, “фирма” или ”частное лицо”
ИНН поставщика	supplier_inn	V(12)	первичный ключ; тип переменной длины используется в связи с тем, что ИНН частного лица состоит из 12 цифр, а фирмы из 10
Адрес	supplier_address	V(100)	многозначное поле

ФИО руководителя	supplier_manager_name	V(50)	обязательное поле, ФИО руководителя совпадает с Названием, если частное лицо
Телефон	supplier_phone	V(30)	многозначное поле
Эл. почта	supplier_email	V(30)	многозначное поле
Условия поставки	supplier_conditions	V(100)	необязательное поле, может указываться: «ежемесячная», «единоразовая» и др.
Дата начала сотрудничества	supplier_start_date	D	обязательное поле
Дата окончания сотрудничества	supplier_end_date	D	необязательное поле; если договор с поставщиком не расторгнут и не закончился, поле остается пустым
Сотрудник, заключивший договор с поставщиком	supplier_employee	C(12)	Обязательный внешний ключ на Staff

В качестве первичного ключа для отношения **Поставщики** будем использовать уникальный атрибут ИНН поставщика

Отношение **Сотрудники (Staff)**:

Таблица 3. Схема отношения **Сотрудники (Staff)**

<i>Содержание поля</i>	<i>Имя поля</i>	<i>Тип, длина</i>	<i>Примечания</i>
Фамилия, имя, отчество	staff_name	V(50)	обязательное поле
Дата рождения	staff_born	D	обязательное поле
Пол	staff_sex	C(1)	обязательное поле, «м» или «ж»
Серия и номер паспорта	staff_passport	C(10)	обязательное поле
ИНН	staff_inn	C(12)	<b>первичный ключ</b>
СНИЛС	staff_snils	C(11)	обязательное уникальное поле
Должность	staff_post	V(30)	обязательное поле
Оклад	staff_salary	N(8,2)	обязательное поле, > МРОТ
Вид образования	staff_education	V(50)	обязательное многозначное поле
Адрес проживания	staff_adress	V(100)	многозначное поле
Телефоны	staff_phone	V(30)	многозначное поле
Электронная почта	staff_email	V(30)	многозначное поле
Начальник	staff_cheif	C(12)	необязательный внешний ключ к Staff

В качестве первичного ключа для отношения **Сотрудники** будем использовать уникальный атрибут ИНН



## Отношение **Покупатели (Buyers)**:

Таблица 4. Схема отношения **Покупатели (Buyers)**

Содержание поля	Имя поля	Тип, длина	Примечания
Идентифицирующий номер	buyer_id	C(10)	<b>суррогатный первичный ключ</b>
ФИО	buyer_name	V(50)	необязательное поле, указывается по необходимости
Телефон	buyer_phone	V(30)	многозначное поле; необязательное поле, указывается по необходимости
Эл. почта	buyer_email	V(30)	многозначное поле; необязательное поле, указывается по необходимости
Адрес доставки	buyer_adress	V(100)	многозначное поле; необязательное поле, указывается по необходимости
Зарегистрировавший сотрудник	buyer_employee	C(12)	обязательный внешний ключ на Staff

В качестве первичного ключа для отношения **Покупатели** будем использовать «искусственный» атрибут Идентифицирующий номер

## Отношение **Товары (Products)**:

Таблица 5. Схема отношения **Товары (Products)**

Содержание поля	Имя поля	Тип, длина	Примечания
Артикул	product_id	C(10)	<b>первичный ключ</b>
Категория	product_category	V(50)	обязательное поле
Наличие	product_sign	C(1)	обязательное поле, «0» или «1»
Цена	product_price	N(8, 2)	обязательное поле
Размер	product_size	C(4)	обязательное поле
Пол	product_sex	C(1)	обязательное поле «м» или «ж» или «у»
Бренд	product_brand	V(30)	необязательное поле
Преобладающий материал	product_material	V(30)	необязательное поле, указывается материал с большей долей состава
Цвет	product_color	V(30)	обязательное поле
Изношенность	product_state	C(1)	обязательное поле, «1», или «2», или «3», или «4», или «5»
Особенности	product_features	V(100)	необязательное поле
Покупатель товара	product_buyer	C(10)	необязательный внешний ключ к Buyers (пока товар не купили, поле остается пустым)
Поставщик товара	product_supplier	V(12)	обязательный внешний ключ к Suppliers

Продавец товара	product_staff	C(12)	необязательный внешний ключ к Staff (пока товар не купили, поле остается пустым)
-----------------	---------------	-------	--

В качестве первичного ключа для отношения **Товары** будем использовать уникальный атрибут Артикул

## Нормализация полученных отношений (до 4НФ)

### Поставщики.

Составной атрибут ФИО руководителя разобьем на два атрибута Фамилия руководителя и Имя и отчество руководителя. Вынесем контактные данные (телефоны и почты) поставщиков в отдельное отношение **Телефоны-почты поставщиков (Suppliers\_phem)**. В этом отношении не будет первичного ключа, так как на него никто не ссылается, и будет обязательный внешний ключ к отношению **Поставщики (Suppliers)**.

По такому же принципу вынесем адреса поставщиков в отдельное отношение **Адреса поставщиков (Suppliers\_addresses)**. Разобьем составной атрибут Адрес на атрибуты: Страна, Регион, Населенный пункт, Улица, Дом, Литера.

Тогда отношение **Поставщики** будет разбито на 3 отношение: **Поставщики**, **Телефоны-почты поставщиков**, **Адреса поставщиков** (см. таблицы 6–8)

Таблица 6. Схема отношения **Поставщики (Suppliers)** после нормализации

<i>Содержание поля</i>	<i>Имя поля</i>	<i>Тип, длина</i>	<i>Примечания</i>
Название (ФИО)	supplier_name	V(50)	обязательное поле, в случае если поставщиком является частное лицо в названии указывается ФИО
Тип	supplier_type	V(12)	обязательное поле, “фирма” или “частное лицо”
ИНН поставщика	supplier_inn	V(12)	<b>первичный ключ</b> ; тип переменной длины используется в связи с тем, что ИНН частного лица состоит из 12 цифр, а фирмы из 10
Фамилия руководителя	supplier_manager_surname	V(25)	обязательное поле, ФИО руководителя совпадает с Названием, если частное лицо
Имя и отчество руководителя	supplier_manager_name	V(30)	обязательное поле, ФИО руководителя совпадает с Названием, если частное лицо

Условия поставки	supplier_conditions	V(100)	необязательное поле, может указываться: «ежемесячная», «единоразовая» и др.
Дата начала сотрудничества	supplier_start_date	D	обязательное поле
Дата окончания сотрудничества	supplier_end_date	D	необязательное поле; если договор с поставщиком не расторгнут и не закончился, поле остается пустым; дата окончания должна быть позже даты начала сотрудничества
Сотрудник, заключивший договор с поставщиком	supplier_staff	C(12)	Обязательный внешний ключ на Staff

Таблица 7. Схема отношения **Телефоны-почты поставщиков (Suppliers\_phem)**

Содержание поля	Имя поля	Тип, длина	Примечания
Идентификатор поставщика	supplier_id	V(12)	обязательный внешний ключ к Suppliers
Телефон	supplier_phone	V(20)	хотя бы одно из этих полей должно быть определено
Электронная почта	supplier_email	V(20)	

Таблица 8. Схема отношения **Адреса поставщиков (Suppliers\_addresses)**

Содержание поля	Имя поля	Тип, длина	Примечания
Идентификатор поставщика	supplier_id	V(12)	обязательный внешний ключ к Suppliers
Страна	supplier_country	V(20)	обязательное поле
Регион	supplier_region	V(20)	обязательное поле
Населенный пункт	supplier_city	V(20)	обязательное поле
Улица	supplier_street	V(20)	обязательное поле
Дом	supplier_house	V(5)	обязательное поле
Литера	supplier_letter	C(1)	необязательное поле

### Сотрудники.

Составной атрибут ФИО также разобьем на два атрибута. Атрибуты Должность и Оклад вынесем в отдельный справочник **Должности (Posts)** с первичным ключом Должность, так как Оклад транзитивен по отношению к Должности. Также заведем справочник **Виды образования (Education)** с суррогатным первичным ключом-идентификатором. Аналогично отношениям **Телефоны-почты поставщиков (Suppliers\_phem)**, **Адреса поставщиков (Suppliers\_addresses)** вынесем контактные данные сотрудников и их адреса в отдельные отношения **Телефоны-почты сотрудников (Staff\_phem)** и **Адреса сотрудников (Staff\_addresses)**.

Таким образом, отношение **Сотрудники** будет разбито на 5 отношений:  
**Сотрудники, Должности, Виды образования, Телефоны-почты сотрудников и Адреса сотрудников** (см таблицы 9–13)

Таблица 9. Схема отношения **Сотрудники (Staff)** после нормализации

Содержание поля	Имя поля	Тип, длина	Примечания
Фамилия	staff_surname	V(25)	обязательное поле
Имя и отчество	staff_name	V(30)	обязательное поле
Дата рождения	staff_born	D	обязательное поле
Пол	staff_sex	C(1)	обязательное поле, 'м' или 'ж'
Серия и номер паспорта	staff_passport	C(10)	обязательное поле
ИНН	staff_inn	C(12)	<b>первичный ключ</b>
СНИЛС	staff_snils	C(11)	обязательное уникальное поле
Должность	staff_post	V(30)	обязательный внешний ключ к Posts
Вид образования	staff_education	N(1)	обязательный внешний ключ к Education
Начальник	staff_chief	C(12)	необязательный внешний ключ к Staff

Таблица 10. Схема отношения **Должности (Posts)**

Содержание поля	Имя поля	Тип, длина	Примечания
Должность	post	V(30)	<b>первичный ключ</b>
Оклад	salary	N(8,2)	обязательное поле, > МРОТ

Таблица 11. Схема отношения **Виды образования (Education)**

Содержание поля	Имя поля	Тип, длина	Примечания
Идентификатор	education_id	N(1)	<b>суррогатный первичный ключ</b>
Название вида образования	education_name	V(20)	обязательное поле

Таблица 12. Схема отношения **Телефоны-почты сотрудников (Staff\_phem)**

Содержание поля	Имя поля	Тип, длина	Примечания
Идентификатор сотрудника	staff_id	C(12)	обязательный внешний ключ к Staff
Телефон	staff_phone	V(20)	хотя бы одно из этих полей должно быть определено
Электронная почта	staff_email	V(20)	

Таблица 13. Схема отношения **Адреса сотрудников (Staff\_addresses)**

Содержание поля	Имя поля	Тип, длина	Примечания
Идентификатор сотрудника	staff_id	C(12)	обязательный внешний ключ к Staff
Страна	staff_country	V(20)	обязательное поле
Регион	staff_region	V(20)	обязательное поле
Населенный пункт	staff_city	V(20)	обязательное поле
Улица	staff_street	V(20)	обязательное поле

Дом	staff_house	V(5)	обязательное поле
Литера	staff_letter	C(1)	необязательное поле
Квартира	staff_flat	V(5)	необязательное поле

### Покупатели.

Составной атрибут ФИО тут также разобьем на два. По аналогии с предыдущими пунктами вынесем контактные данные и адреса покупателей в отдельные отношения **Телефоны-почты покупателей (Buyers\_phem)** и **Адреса покупателей (Buyers\_addresses)**

Отношение **Покупатели** разобьётся на 3 отдельных отношения: **Покупатели**, **Адреса покупателей**, **Телефоны-почты покупателей** (см таблицы 14–16)

Таблица 14. Схема отношения **Покупатели (Buyers)** после нормализации

Содержание поля	Имя поля	Тип, длина	Примечания
Идентифицирующий номер	buyer_id	C(10)	<b>суррогатный первичный ключ</b>
Фамилия	buyer_surname	V(25)	необязательное поле, указывается по необходимости
Имя и отчество	buyer_name	V(30)	необязательное поле, указывается по необходимости
Зарегистрировавший сотрудник	buyer_staff	C(12)	обязательный внешний ключ на Staff

Таблица 15. Схема отношения **Адреса покупателей (Buyers\_addresses)**

Содержание поля	Имя поля	Тип, длина	Примечания
Идентификатор покупателя	buyer_id	C(10)	обязательный внешний ключ к Buyers
Страна	buyer_country	V(20)	обязательное поле
Регион	buyer_region	V(20)	обязательное поле
Населенный пункт	buyer_city	V(20)	обязательное поле
Улица	buyer_street	V(20)	обязательное поле
Дом	buyer_house	V(5)	обязательное поле
Литера	buyer_letter	C(1)	необязательное поле
Квартира	buyer_flat	V(5)	необязательное поле

Таблица 16. Схема отношения **Телефоны-почты покупателей (Buyers\_phem)**

Содержание поля	Имя поля	Тип, длина	Примечания
Идентификатор покупателя	buyer_id	C(10)	обязательный внешний ключ к buyers
Телефон	buyer_phone	V(20)	хотя бы одно из этих полей должно быть определено

### Товары.

Выделим в качестве отдельных справочников характеристика товара: **Категории (Categories)**, **Бренды (Brands)**, **Материалы (Materials)**, **Цвета (Colors)**. В каждом из них в качестве суррогатного первичного ключа – численный идентификатор.

В итоге отношение Товары разобьется на 5 отношений (см таблицы 17–21)

Таблица 17. Схема отношения **Товары (Products)** после нормализации

<i>Содержание поля</i>	<i>Имя поля</i>	<i>Тип, длина</i>	<i>Примечания</i>
Артикул	product_id	C(10)	<b>первичный ключ</b>
Категория	product_category	N(3)	обязательный внешний ключ на Categories
Наличие	product_sign	C(1)	обязательное поле, «0» или «1»
Цена	product_price	N(8, 2)	обязательное поле, > 0
Размер	product_size	V(5)	обязательное поле
Пол	product_sex	C(1)	обязательное поле «м» или «ж» или «у»
Бренд	product_brand	N(4)	необязательный внешний ключ на Brands
Материал	product_material	N(3)	необязательный внешний ключ на Materials
Цвет	product_color	N(3)	обязательный внешний ключ на Colors
Изношенность	product_state	N(1)	обязательное поле, «1» или «2» или «3» или «4» или «5»
Особенности	product_features	V(100)	необязательное поле
Покупатель товара	product_buyer	C(10)	необязательный внешний ключ к Buyers (пока товар не купили, поле остается пустым)
Поставщик товара	product_supplier	V(12)	обязательный внешний ключ к Suppliers
Продавец товара	product_staff	C(12)	необязательный внешний ключ к Staff (пока товар не купили, поле остается пустым)

Таблица 18. Схема отношения **Категории (Categories)**

<i>Содержание поля</i>	<i>Имя поля</i>	<i>Тип, длина</i>	<i>Примечания</i>
Идентификатор	category_id	N(3)	<b>первичный ключ</b>
Категория	category	V(50)	обязательное поле

Таблица 19. Схема отношения **Бренды (Brands)**

<i>Содержание поля</i>	<i>Имя поля</i>	<i>Тип, длина</i>	<i>Примечания</i>
Идентификатор	brand_id	N(4)	<b>первичный ключ</b>
Бренд	brand	V(30)	обязательное поле

Таблица 20. Схема отношения **Материалы (Materials)**

<i>Содержание поля</i>	<i>Имя поля</i>	<i>Тип, длина</i>	<i>Примечания</i>
Идентификатор	material_id	N(3)	первичный ключ
Материал	material	V(30)	обязательное поле, указывается материал с большей долей от общего состава

Таблица 21. Схема отношения **Цвета (Colors)**

Содержание поля	Имя поля	Тип, длина	Примечания
Идентификатор	color_id	N(3)	<b>первичный ключ</b>
Цвета	color	V(30)	обязательное поле

Схема базы данных после нормализации (см. рисунок 3)

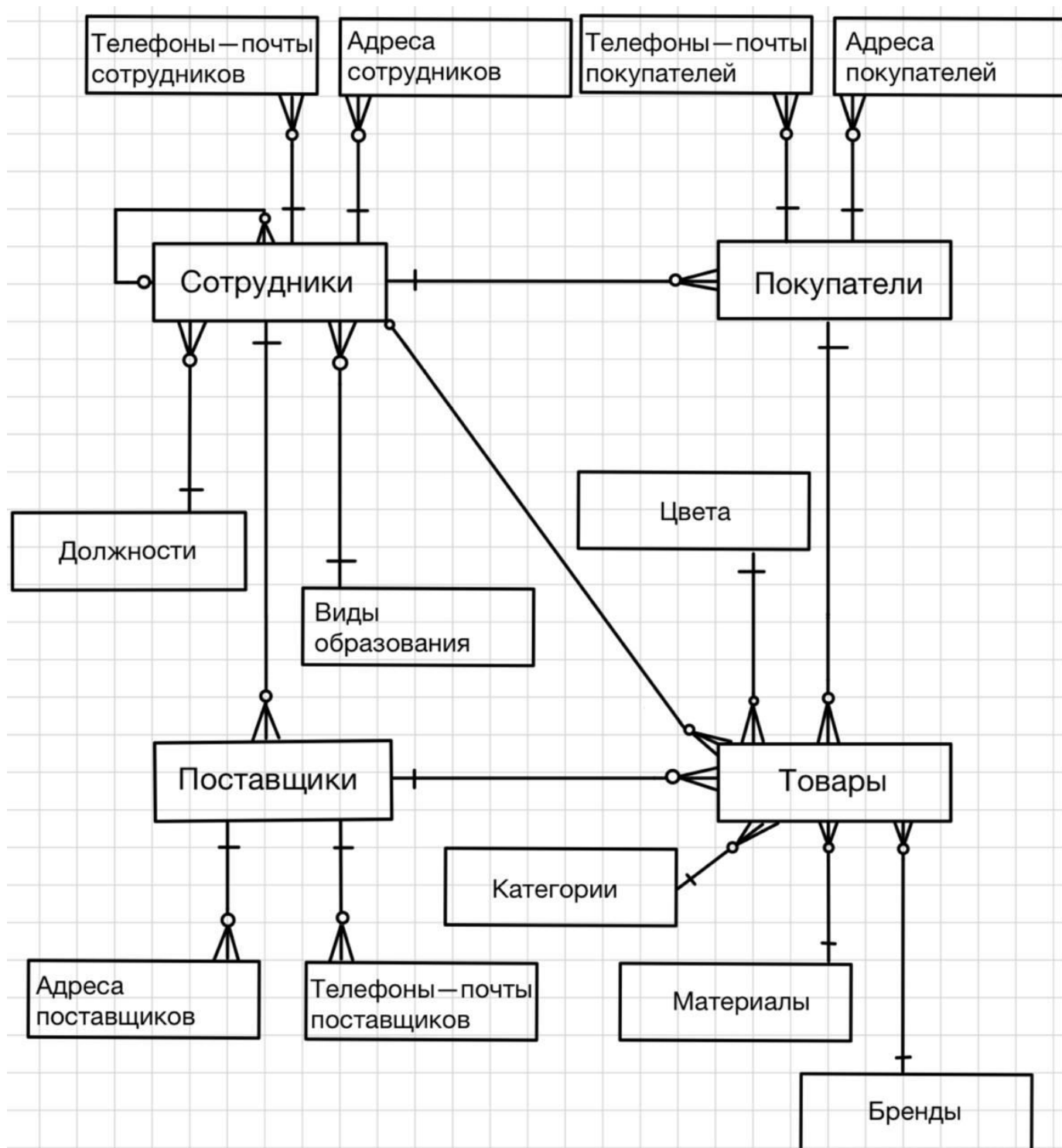


Рисунок 3. Окончательная схема БД после нормализации

## Определение дополнительных ограничений целостности

Перечислим ограничения целостности, которые не указаны ранее.

1. Дата начала сотрудничества (отношение Поставщики) не больше текущей даты.
2. Поля, содержащие ИНН, СНИЛС или номер паспорта могут включать только цифры.
3. Нумерация в поле Идентификатор (отношения Цвета) начинается с 0 (за нулем закреплено неопределенное значение параметра (в данном случае «разноцветный»)) и является непрерывной.
4. Нумерация в поле Идентификатор (отношений Виды образования, Категории, Бренды, Материалы) начинается с 1 (так как нет неопределенных значений, нумерация не с нуля) и является непрерывной.
5. Сотрудники с должностями «Администратор», «Грузчик», «Уборщик» не могут зарегистрировать покупателя.
6. Сотрудники с должностями «Консультант-кассир», «Грузчик» или «Уборщик» не могут заключать договора с поставщиками.

Ограничения 1–2 можно реализовать в схеме отношения. Ограничения 3–4 нельзя гарантировать даже с помощью триггера или программы, можно только проверить, что это правило соблюдается и выдать сообщение о его нарушении. Ограничения 5–6 можно учесть с помощью триггеров.

## Описание групп пользователей и прав доступа

Опишем для каждой группы пользователей права доступа к каждой таблице. Права доступа должны быть распределены так, чтобы для каждого объекта БД был хотя бы один пользователь, который имеет право добавлять и удалять данные из объекта. Ниже описаны используемые сокращения (Таблица 22) и распределение прав доступа по должностям (Таблица 23).

Таблица 22. Обозначения прав доступа

Обозначение	Расшифровка	Альтернативное обозначение
I (insert)	добавление данных	C (create)
S (select)	чтение данных	R (read)
U (update)	модификация данных	U (update)
D (delete)	удаление данных	D (delete)

Таблица 23. Права доступа к таблицам для групп пользователей

Таблицы	Администратор	Консультант-кассир	Грузчик	Уборщик
Поставщики	SUID	S	S	
Телефоны-почты поставщиков	SUID	S	S	
Адреса поставщиков	SUID	S	S	
Сотрудники	SUID	S	S	S
Должности	SUID			
Виды образования	SUID			



Телефоны-почты сотрудников	SUID	S	S	S
Адреса сотрудников	SUID			
Покупатели	SUID	SUI	S	
Адреса покупателей	SUID	SUI	S	
Телефоны-почты покупателей	SUID	SUI	S	
Товары	SUID	SUID		S
Категории	SUID	S		
Бренды	SUID	S		
Материалы	SUID	S		
Цвета	SUID	S		

## Реализация проекта базы данных

### Создание таблиц

#### 1) Отношение Posts (должности):

```
create table Posts (
    post varchar(30) primary key,
    salary numeric(8, 2) not null,
    constraint check_salary check(salary >= 19000)
);
-- 19000 - размер МРОТ на момент создания таблицы.
```

#### 2) Отношение Education (виды образования):

```
create table Education (
    education_id numeric(1) primary key,
    education_name varchar(20) not null
);
```

#### 3) Отношение Colors (цвета):

```
create table Colors (
    color_id numeric(3) primary key,
    color varchar(30) not null
);
```

#### 4) Отношение Categories (категории):

```
create table Categories (
    category_id numeric(3) primary key,
    category varchar(50) not null
);
```

#### 5) Отношение Materials (материалы):

```
create table Materials (
    material_id numeric(3) primary key,
    material varchar(30) not null
);
```

**6) Отношение Brands (бренды):**

```
create table Brands (  
    brand_id numeric(4) primary key,  
    brand varchar(30) not null  
);
```

**7) Отношение Staff (сотрудники):**

```
create table Staff (  
    staff_surname varchar(25) not null,  
    staff_name varchar(30) not null,  
    staff_born date not null,  
    staff_sex char(1) not null constraint check_sex check(  
        staff_sex in ('ж', 'м')),  
    staff_passport char(10) not null constraint uniq_pasp unique  
constraint check_pasp check(  
    cast(staff_passport as int)> 100000000),  
    staff_inn char(12) primary key constraint uniq_inn unique  
constraint check_inn check(  
    cast(staff_inn as int)> 100000000000),  
    staff_snils char(11) not null constraint uniq_snils unique  
constraint check_snils check(  
    cast(staff_snils as int)> 10000000000),  
    staff_post varchar(30) not null constraint fk_staff_posts  
references Posts,  
    staff_education numeric(1) not null constraint fk_edu_posts  
references Education,  
    staff_chief char(12) references Staff  
);
```

**8) Отношение Staff\_phem (телефоны-почты сотрудников):**

```
create table Staff_phem (  
    staff_id char(12) not null constraint fk_phem_staff references  
Staff,  
    staff_phone varchar(20),  
    staff_email varchar(20),  
    constraint check_phem check(  
        staff_phone is not null  
        or staff_email is not null)  
);
```

**9) Отношение Staff\_addresses (адреса сотрудников):**

```
create table Staff_addresses (  
    staff_id char(12) not null constraint fk_phem_staff references  
Staff,  
    staff_country varchar(20) not null,  
    staff_region varchar(20) not null,  
    staff_city varchar(20) not null,  
    staff_street varchar(20) not null,  
    staff_house varchar(5) not null,  
    staff_letter char(1),  
    staff_flat varchar(5)  
);
```

#### 10) Отношение Suppliers (поставщики):

```
create table Suppliers (  
    supplier_name varchar(50) not null,  
    supplier_type varchar(12) not null constraint check_type check(  
        supplier_type in ('фирма', 'частное лицо')),  
    supplier_inn varchar(12) primary key constraint uniq_suppl_inn  
unique constraint check_inn check(  
    cast(supplier_inn as int)> 1000000000),  
    supplier_manager_surname varchar(25) not null,  
    supplier_manager_name varchar(30) not null,  
    supplier_conditions varchar(100),  
    supplier_start_date date not null,  
    supplier_end_date date constraint check_end_date check(  
        supplier_end_date > supplier_start_date),  
    supplier_staff char(12) not null constraint fk_supplier_staff  
references Staff  
);
```

#### 11) Отношение Suppliers\_phem (телефоны-почты поставщиков):

```
create table Suppliers_phem (  
    supplier_id char(12) not null constraint fk_phem_supplier  
references Suppliers,  
    supplier_phone varchar(20),  
    supplier_email varchar(20),  
    constraint check_phem check(  
        supplier_phone is not null  
        or supplier_email is not null)  
);
```

#### 12) Отношение Suppliers\_adresses (адреса поставщиков):

```
create table Suppliers_adresses (  
    supplier_id char(12) not null constraint fk_phem_supplier  
references Suppliers,  
    supplier_country varchar(20) not null,  
    supplier_region varchar(20) not null,  
    supplier_city varchar(20) not null,  
    supplier_street varchar(20) not null,  
    supplier_house varchar(5) not null,  
    supplier_letter char(1)  
);
```

#### 13) Отношение Buyers (покупатели):

```
create table Buyers (  
    buyer_id char(10) primary key,  
    buyer_surname varchar(25),  
    buyer_name varchar(30),  
    buyer_staff char(12) not null constraint fk_buyer_staff  
references Staff  
);
```

#### 14) Отношение Buyers\_phem (телефоны-почты покупателей):

```
create table Buyers_phem (  

```

```

    buyer_id char(10) not null constraint fk_phem_buyer references
Buyers,
    buyer_phone varchar(20),
    buyer_email varchar(20),
    constraint check_phem check(
        buyer_phone is not null
        or buyer_email is not null)
);

```

#### 15) Отношение Buyers\_adresses (адреса покупателей):

```

create table Buyers_adresses (
    buyer_id char(10) not null constraint fk_phem_staff references
Buyers,
    buyer_country varchar(20) not null,
    buyer_region varchar(20) not null,
    buyer_city varchar(20) not null,
    buyer_street varchar(20) not null,
    buyer_house varchar(5) not null,
    buyer_letter char(1),
    buyer_flat varchar(5)
);

```

#### 16) Отношение Products (товары):

```

create table Products (
    product_id char(10) primary key,
    product_category numeric(3) not null constraint
fk_product_category references Categories,
    product_sign char(1) not null constraint check_sign check(
        product_sign in ('1', '0')),
    product_price numeric(8, 2) not null constraint check_price
check(product_price > 0),
    product_size varchar(5) not null,
    product_sex char(1) not null constraint check_sex check(
        product_sex in ('м', 'ж', 'y')),
    product_brand numeric(4) references Brands,
    product_material numeric(3) references Materials,
    product_color numeric(3) not null constraint fk_product_color
references Colors,
    product_state numeric(1) not null constraint check_state check(
        product_state >= 1
        AND product_state <= 5),
    product_features varchar(100),
    product_buyer char(10) references Buyers,
    product_supplier varchar(12) not null constraint
fk_product_supplier references Suppliers,
    product_staff char(12) references Staff
);

```

Начальный состав данных для **справочных таблиц** (начальный состав указан только для справочных таблиц, так как содержание других таблиц не входит в проект БД и будет добавляться по мере поступления данных):

#### 1) Education (виды образования):

```

insert into Education values (1, 'начальное');
insert into Education values (2, 'среднее');
insert into Education values (3, 'средне-специальное');
insert into Education values (4, 'высшее');

```

## 2) Таблица Posts (должности):

```

insert into Posts values ('Администратор', 70360.87);
insert into Posts values ('Консультант-кассир', 62520.09);
insert into Posts values ('Грузчик', 56789.80);
insert into Posts values ('Уборщик', 51250.76);

```

## 3) Таблица Colors (цвета):

```

insert into Colors values (0, 'разноцветный');
insert into Colors values (1, 'красный');
insert into Colors values (2, 'синий');
insert into Colors values (3, 'зеленый');
insert into Colors values (4, 'желтый');
insert into Colors values (5, 'оранжевый');
insert into Colors values (6, 'фиолетовый');
insert into Colors values (7, 'розовый');
insert into Colors values (8, 'черный');
insert into Colors values (9, 'белый');
insert into Colors values (10, 'серый');
insert into Colors values (11, 'коричневый');
insert into Colors values (12, 'голубой');
insert into Colors values (13, 'кремовый');

```

## 4) Таблица Categories (категории):

```

insert into Categories values (1, 'футболки');
insert into Categories values (2, 'джинсы');
insert into Categories values (3, 'платья');
insert into Categories values (4, 'юбки');
insert into Categories values (5, 'рубашки');
insert into Categories values (6, 'куртки');
insert into Categories values (7, 'брюки');
insert into Categories values (8, 'пиджаки');
insert into Categories values (9, 'пальто');
insert into Categories values (10, 'шорты');
insert into Categories values (11, 'туфли');
insert into Categories values (12, 'ботинки');
insert into Categories values (13, 'кроссовки');
insert into Categories values (14, 'сандалии');
insert into Categories values (15, 'сапоги');

```

## 5) Таблица Materials (материалы):

```

insert into Materials values (1, 'хлопок');
insert into Materials values (2, 'акрил');
insert into Materials values (3, 'шерсть');
insert into Materials values (4, 'кожа');
insert into Materials values (5, 'полиэстер');
insert into Materials values (6, 'нейлон');
insert into Materials values (7, 'шелк');
insert into Materials values (8, 'лен');
insert into Materials values (9, 'вельвет');

```

```

insert into Materials values (10, 'твид');
insert into Materials values (11, 'деним');
insert into Materials values (12, 'велюр');
insert into Materials values (13, 'спандекс');
insert into Materials values (14, 'кашемир');
insert into Materials values (15, 'шифон');
insert into Materials values (16, 'полиуретан');
insert into Materials values (17, 'эластан');
insert into Materials values (18, 'акрил');
insert into Materials values (19, 'замша');

```

## 6) Таблица Brands (бренды):

```

insert into Brands values (1, 'Nike');
insert into Brands values (2, 'Adidas');
insert into Brands values (3, 'Zara');
insert into Brands values (4, 'H&M');
insert into Brands values (5, 'Gucci');
insert into Brands values (6, 'Levis');
insert into Brands values (7, 'Prada');
insert into Brands values (8, 'Balenciaga');
insert into Brands values (9, 'Versace');
insert into Brands values (10, 'Calvin Klein');
insert into Brands values (11, 'Puma');
insert into Brands values (12, 'Converse');
insert into Brands values (13, 'New Balance');
insert into Brands values (14, 'Louis Vuitton');
insert into Brands values (15, 'Fendi');
insert into Brands values (16, 'Under Armour');
insert into Brands values (17, 'Tommy Hilfiger');
insert into Brands values (18, 'Ralph Lauren');
insert into Brands values (19, 'Chanel');
insert into Brands values (20, 'Burberry');

```

## Создание представлений (готовых запросов)

Представления являются хранимыми запросами и не содержат данных. Они необходимы для того, чтобы пользователь мог быстро извлечь данные в удобном для себя виде. В ходе анализа ПрО была определена потребность в наличии следующих представлений:

### 1) Контактные данные сотрудников:

```

CREATE VIEW Contact_Details AS
SELECT s.staff_inn AS "staff_INN",
       s.staff_surname || ' ' || s.staff_name AS "staff_full_name",
       sp.staff_phone AS "staff_phone",
       sp.staff_email AS "staff_email"
FROM Staff s
LEFT JOIN Staff_phem sp ON s.staff_inn = sp.staff_id;

```

### 2) Оклад сотрудников:

```

CREATE VIEW Salary_Details AS
SELECT s.staff_inn AS "staff_INN",
       s.staff_surname || ' ' || s.staff_name AS "staff_full_name",

```

```

        s.staff_post AS "staff_post",
        p.salary AS "staff_salary"
FROM Staff s
JOIN Posts p ON s.staff_post = p.post;

```

### 3) Сотрудники и покупатели, которых они зарегистрировали:

```

CREATE VIEW Staff_Buyers AS
SELECT s.staff_inn AS "staff_INN",
       s.staff_surname || ' ' || s.staff_name AS "staff_full_name",
       (SELECT string_agg(b.buyer_id, ', ')
        FROM Buyers b
        WHERE b.buyer_staff = s.staff_inn) AS "staff_buyers"
FROM Staff s;

```

### 4) Поставщики, с которыми в настоящее время заключен контракт (поле с датой окончания сотрудничества пусто) с их адресом и контактными данными:

```

CREATE VIEW Active_Suppliers AS
SELECT s.supplier_name AS "supplier_name",
       s.supplier_inn AS "supplier_inn",
       sa.supplier_country || ', ' || sa.supplier_region || ', ' ||
sa.supplier_city || ', ' || sa.supplier_street || ' ' ||
sa.supplier_house || COALESCE(' ' || sa.supplier_letter, '') AS
"supplier_adress",
       sp.supplier_phone AS "supplier_phone",
       sp.supplier_email AS "supplier_email"
FROM Suppliers s
JOIN Suppliers_phem sp ON s.supplier_inn = sp.supplier_id
JOIN Suppliers_adresses sa ON s.supplier_inn = sa.supplier_id
WHERE s.supplier_end_date IS NULL;

```

### 5) Сотрудники и поставщики, с которыми они заключили договоры:

```

CREATE VIEW Staff_Supplier_Contracts AS
SELECT s.staff_inn AS "staff_inn",
       s.staff_surname || ' ' || s.staff_name AS "staff_full_name",
       string_agg(sp.supplier_inn, ', ') AS "staff_suppliers"
FROM Staff s
JOIN Suppliers sp ON s.staff_inn = sp.supplier_staff
GROUP BY s.staff_inn, s.staff_surname, s.staff_name;

```

### 6) Товары в наличии (значение «0» в поле product\_sign):

```

CREATE VIEW Available_Products AS
SELECT p.product_id,
       p.product_price,
       p.product_size,
       p.product_sex,
       p.product_features,
       b.brand AS "product_brand",
       m.material AS "product_material",
       c.category AS "product_category",
       co.color AS "product_color",
       p.product_state
FROM Products p
JOIN Brands b ON p.product_brand = b.brand_id
JOIN Materials m ON p.product_material = m.material_id

```

```
JOIN Categories c ON p.product_category = c.category_id
JOIN Colors co ON p.product_color = co.color_id
WHERE p.product_sign = '1';
```

Для работы с этими представлениями соответствующим пользователям нужно определить права доступа к представлениям (Таблица 24).

Таблица 24. Права доступа к представлениям для групп пользователей

Представления	Администратор	Консультант-кассир	Грузчик	Уборщик
Контактные данные сотрудников	S	S	S	S
Оклад сотрудников	S			
Сотрудники и покупатели, которых они зарегистрировали	S	S		
Поставщики, с которыми в настоящее время заключен контракт	S		S	
Сотрудники и поставщики, с которыми они заключили договоры	S			
Товары в наличии	S	S		

## Назначение прав доступа

Создадим следующие роли: admin (администратор), cashier (консультант-кассир), loader (грузчик) и cleaner (уборщик).

Права доступа пользователей предоставляются с помощью команды GRANT. Рассмотрим для примера права консультанта-кассира. Права доступа к отношениям Suppliers (поставщики), Staff (сотрудники), Buyers (покупатели) и Products (товары) могут быть описаны следующим образом:

```
grant select on Suppliers to cashier;
grant select,update,insert on Buyers to cashier;
grant ALL on Products to cashier;
```

Права доступа администратора к представлению “Оклад сотрудников” могут быть описаны следующим образом:

```
grant select on Salary_Details to admin;
```

Если сотрудник не является администратором, он не получит данных через это представление и не сможет воспользоваться правами доступа к нему.

Права доступа грузчика к представлению “Поставщики, с которыми в настоящее время заключен контракт” могут быть описаны следующим образом:



```
grant select on Active_Suppliers to loader;
```

Если сотрудник не является грузчиком или администратором, он не получит данных через это представление и не сможет воспользоваться правами доступа к нему.

Права доступа уборщика к представлению “Контактные данные сотрудников” могут быть описаны следующим образом:

```
grant select on Contact_Details to cleaner;
```

## Создание триггеров

Учтем, что сотрудник с должностью «Администратор», «Грузчик», «Уборщик» не может регистрировать покупателя, то есть его ИНН не может быть в поле buyer\_staff отношения **Покупатели**.

Реализуем эту проверку с помощью триггера, создав сначала триггерную функцию:

```
CREATE OR REPLACE FUNCTION check_staff_post()
RETURNS TRIGGER AS $$
BEGIN
    IF EXISTS (
        SELECT 1
        FROM Staff
        WHERE staff_inn = NEW.buyer_staff
        AND staff_post IN ('Администратор', 'Грузчик', 'Уборщик')
    ) THEN
        RAISE EXCEPTION 'Сотрудник с должностью "%", "%", или "%" не
        может быть зарегистрирован как покупатель', 'Администратор',
        'Грузчик', 'Уборщик';
    END IF;
    RETURN NEW;
END;
$$ LANGUAGE plpgsql;
```

После создания триггерной функции можно создать сам триггер:

```
CREATE TRIGGER prevent_staff_post_buyer
BEFORE INSERT ON Buyers
FOR EACH ROW
EXECUTE FUNCTION check_staff_post();
```

Также учтем, что сотрудник с должностью «Консультант-кассир», «Грузчик» или «Уборщик» не может заключать договора с поставщиками:

```
CREATE OR REPLACE FUNCTION check_staff_post_supplier()
RETURNS TRIGGER AS $$
BEGIN
    IF EXISTS (
        SELECT 1
        FROM Staff
        WHERE staff_inn = NEW.supplier_staff
        AND staff_post IN ('Консультант-кассир', 'Грузчик', 'Уборщик')
    ) THEN
        RAISE EXCEPTION 'Сотрудник с должностью "%", "%", или "%" не
        может заключать договоры с поставщиками', 'Консультант-кассир',
        'Грузчик', 'Уборщик';
    END IF;
END;
```

```

RETURN NEW;
END;
$$ LANGUAGE plpgsql;

CREATE TRIGGER prevent_staff_post_supplier
BEFORE INSERT ON Suppliers
FOR EACH ROW
EXECUTE FUNCTION check_staff_post_supplier();

```

## Создание индексов

Для повышения эффективности работы с данными необходимо создать индексы для всех внешних ключей. Приведём примеры создания индексов:

```

create index ind_sup_staff on Suppliers(supplier_staff);
create index ind_buyer_staff on Buyers(buyer_staff);
create index ind_prod_cat on Products(product_category);
create index ind_sup_phem_id on Suppliers_phem(supplier_id);

```

Полезными окажутся составные индексы по ФИО покупателя в отношении Buyers (покупатели), ФИО руководителя в отношении Suppliers (поставщики). Также нужен составной индекс по названию и дате начала сотрудничества в отношении Suppliers (поставщики) для ускорения поиска поставщиков. Создадим эти индексы:

```

create index ind_buyer_name on Buyers(buyer_surname, buyer_name);
create index ind_sup_man_name on
Suppliers(supplier_manager_surname, supplier_manager_name);
create index ind_sup_name_date on Suppliers(supplier_name,
supplier_name);

```

Индексы по таблице Staff (сотрудники) создавать не будем, так как она небольшая.

## Разработка стратегии резервного копирования

Для базы данных секунд-хенд магазина с низкой интенсивностью обновлений и небольшими требованиями к надежности данных, стратегия резервного копирования может быть следующей:

### - Полное резервное копирование ежедневно:

Проводить полное резервное копирование базы данных раз в день, перед окончанием рабочего дня. Это гарантирует сохранность всех данных на момент окончания дня.

### - Автоматическое выполнение перед выключением сервера:

Создать задание в операционной системе, которое автоматически запускается перед выключением сервера. Это позволит автоматически выполнять полное резервное копирование базы данных перед выключением сервера.