|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | | | |
| Федеральное государственное бюджетное  образовательное учреждение высшего образования «Новосибирский государственный технический университет» | | | | |
|  | | | | |
| Кафедра теоретической и прикладной информатики | | | | |
| Пояснительная записка к проекту | | | | |
| по дисциплине «Базы данных и экспертные системы» | | | | |
| **Проектирование и разработка информационных систем и** **баз данных** | | | | |
|  | | | | |
|  | ПМ-84 | ермолаева ярослава | |
| ПМ-84 | чевлытко екатерина | |
| ПМ-85 | Бариев родион | |
| ПМ-85 | гончаренко никита | |
| ПМ-85 | малыгин артём | |
|  | |  | |
| Преподаватель | | Стасышин владимир михайлович | |
|  | |  | |
| Новосибирск, 2021 | | | | |

[1. Анализ предметной области 3](#_Toc90238379)

[Задание 3](#_Toc90238380)

[Виды запросов в информационной системе 4](#_Toc90238381)

[Основные понятия информационной системы 5](#_Toc90238382)

[Структура организации 7](#_Toc90238383)

[Бизнес-процессы торговой компании 9](#_Toc90238384)

[2. Инфологическое проектирование 16](#_Toc90238385)

[Список сущностей предметной области и их атрибуты 17](#_Toc90238386)

[ER-диаграмма 23](#_Toc90238387)

[3. Логическое проектирование 24](#_Toc90238388)

[Скрипт для создания базы данных 30](#_Toc90238389)

[4. Проектирование приложений 35](#_Toc90238390)

[Цели, задачи и требования 35](#_Toc90238391)

[Состав подсистем разрабатываемой ИС, назначение и функции каждого приложения ИС 36](#_Toc90238392)

[Категории пользователей, работающих с приложениями ИС, выполняемые ими функции, права пользователей, категории данных, с которыми они работают 38](#_Toc90238393)

[Отчеты 40](#_Toc90238394)

[Запросы для формирования отчетов 40](#_Toc90238395)

1. Анализ предметной области

Задание

Торговая компания работает в торговых точках разных типов: супермаркеты, магазины, киоски и т. д., в штате которых трудятся продавцы. Торговая компания может иметь в собственности супермаркет и/или магазин, а может арендовать для них отдельные помещения (секции). Супермаркеты, магазины, киоски могут иметь такие характеристики, как размер торговой точки, платежи за аренду, коммунальные услуги. Кроме того, в супермаркетах и магазинах учет проданных товаров ведется персонифицировано с фиксацией имен и характеристик покупателя, чего в киосках сделать невозможно.

Заказы поставщику составляются на основе заявок, поступающих из торговых точек. На основе заявок менеджеры торговой компании выбирают поставщика, формируют заказы, в которых перечисляются наименования товаров и их количество (оно может отличаться от запроса из торговой точки). Если указанный товар ранее не поставлялся, то его наименование пополняет справочник номенклатуры товаров. Рынок поставщиков постоянно изучается, поэтому могут появляться новые поставщики и исчезать старые. При этом одни и те же товары торговая компания может получать от разных поставщиков и, естественно, по различным ценам.

Поступившие товары распределяются по торговым точкам и в любой момент можно получить такое распределение.

Продавцы торговых точек продают товары, учитывая все совершенные продажи, фиксируя номенклатуру и количество проданного товара, а продавцы супермаркетов и магазинов дополнительно фиксируют имена и характеристики покупателей, что позволяет вести учет покупателей и сделанных ими покупок. Торговые точки вправе менять цены на товары в зависимости от спроса и предложения.

Виды запросов в информационной системе

1. Получить перечень и общее число поставщиков указанного вида товара либо некоторого товара в объеме не менее заданного за весь период сотрудничества либо за указанный период.
2. Получить перечень и общее число покупателей указанного вида товара за некоторый период либо сделавших покупку товара в объеме не менее заданного.
3. Получить номенклатуру и объем товаров в указанной торговой точке.
4. Получить актуальные сведения об объеме и ценах на указанный товар по всем торговым точкам, по торговым точкам заданного типа, по конкретной торговой точке.
5. Получить данные о выработке на одного продавца за указанный период по всем торговым точкам, по торговым точкам заданного типа.
6. Получить данные о выработке отдельно взятого продавца определенной торговой точки за указанный период.
7. Получить данные об объеме продаж указанного товара за некоторый период по всем торговым точкам, по торговым точкам заданного типа, по конкретной торговой точке.
8. Получить данные о заработной плате продавцов по всем торговым точкам, по торговым точкам заданного типа, по конкретной торговой точке.
9. Получить сведения о поставках определенного товара указанным поставщиком за все время поставок либо за некоторый период.
10. Получить данные об арендованных помещениях, о количестве и перечне сотрудников, работающих в помещениях указанного типа, в конкретном помещении, по всем арендованным помещениям.
11. Получить данные о рентабельности торговой точки: соотношение объема продаж к накладным расходам (суммарная заработная плата продавцов + платежи за аренду, коммунальные услуги) за указанный период.
12. Получить сведения о поставках товаров по указанному номеру заказа.
13. Получить сведения о покупателях указанного товара за обозначенный (или за весь) период по всем торговым точкам, по торговым точкам указанного типа, по данной торговой точке.
14. Получить сведения о наиболее активных покупателях по всем торговым точкам, по торговым точкам указанного типа, по данной торговой точке.
15. Получить данные о товарообороте торговой точки либо всех торговых точек определенной группы за указанный период.

Основные понятия информационной системы

1. **Сотрудники** 
   1. Старший продавец – создает заявки, принимает товары на торговой точке, подписывает акты о приемки товара.
   2. Продавец – реализует товар, выкладывает товар на прилавки.
   3. Уборщик – следит за чистотой на торговой точке.
   4. Директор – отвечает за согласование открытия/закрытия торговых точек и складов, подписывает все документы.
   5. Рекрутер – отвечает за подбор персонала.
   6. Бухгалтер – ведет финансовую отчетность компании.
   7. Менеджер – заключает договора с поставщиками, обрабатывает заявки торговых точек.

При поступлении сотрудника на работу фиксируются его личные данные и место работы. При повышении сотрудника фиксируются даты ухода и вступления в должность, и они должны совпадать. При увольнении сотрудника фиксируется дата увольнения.

1. **Товары**

Новые товары появляются во время договоренности с поставщиком, при принятии решения о закупке товара, фиксируется название товара, к какому типу он относится, номенклатурный номер (штрих-код), наименование, тип товара (название типа, характеристика типа товара), срок хранения, описание товара.

1. **Поставщики**

Их вносят в базу данных менеджеры после заключения договора о поставке, перед первым заказом товаров. В базе фиксируются наименование поставщика, даты начала и окончания сотрудничества, банковские реквизиты, контактная информация, географический адрес производства. Если сотрудничество прекращается, то фиксируется причина. Периодически, по мере необходимости, фиксируются изменения атрибутов поставщика.

1. **Поставка**

Содержит информацию о том, какой поставщик какой товар поставляет. При этом указываются оптовая цена, максимальный объем поставки, минимальный объем товара в заказе, дата производства товара.

1. **Склад**

Место, в которое поставщики привозят товары. Указываются даты открытия/закрытия склада, географический адрес и площадь.

1. **Заказ**

Список поставок, оформленных менеджером в рамках одного соглашения. По итогу выполнения заказа товары из поставок попадают на склад.

1. **Распределение**

Информация о том, когда и в каком количестве какая поставка на какой склад была исполнена.

1. **Заявка**

Список товаров, необходимых торговой точке. Оформляется старшим продавцом. По итогу выполнения заявки товары со склада попадают в торговую точку.

1. **Поступление**

Информация о том, когда и в каком количестве какой товар был привезен со склада. Также указывается розничная цена.

1. **Цена**

Информация о цене с указанием периода действия цены и признаком актуальности действия.

1. **Торговая точка**

Торговые точки это места реализации товаров торговой компании. Решения об открытии и закрытии торговых точек принимает директор торговой компании. У каждой торговой точки указываются адрес, тип (супермаркет/гипермаркет/киоск), дата открытия и дата закрытия в случае закрытия, а также признак существования торговой точки.

1. **Реализация**

Учет проданных товаров. При продаже товаров фиксируются данные о том, в какой торговой точке была совершена продажа, какой товар был продан, какой сотрудник торговой точки продал товар и какой клиент приобрел товар. Также указываются дата и время продажи и количество проданного товара.

1. **Клиент**

Сведения о покупателях фиксируются при получении ими дисконтных карт. Желающие получить дисконтные карты сообщают свои личные данные (ФИО, дата рождения) продавцу, который оформляет карту. При этом фиксируются дата выдачи карточки и срок ее действия. При совершении первой покупки по карте фиксируется дата активации карточки.

1. **Платеж**

Учет финансовых операций, связанных с расходами торговых точек. При совершении платежа фиксируется его тип (аренда/коммунальные платежи/зарплаты), дата и время совершения платежа, сумма.

Структура организации

1. **Дирекция**

Принимает решение о создании/ликвидации торговых точек, складов, рабочих мест. Контролирует работу остальных подразделений (утверждает решения, определяет обязанности сотрудников).

*Входная информация:*

* Аналитические и финансовые отчеты (от экономического отдела).
* Рекомендации (от экономического отдела).

*Выходная информация:*

* Список торговых точек (во все отделы).
* Количество рабочих мест и требования к персоналу (в отдел кадров).
* Должностные обязанности сотрудников (во все отделы).
* Решения по рекомендациям (в экономический отдел).

1. **Экономический отдел**

Анализирует результаты сотрудничества с поставщиками и ищет новых поставщиков, проводит с ними переговоры о сотрудничестве. Вырабатывает рекомендации по привлечению новых поставщиков по прекращению сотрудничества с поставщиками.

*Входная информация:*

* Сведения о продажах (от торговых отделов).
* Сведения о поставках (от отдела поставок).
* Сведения о состоянии рынка (цены, отчеты, опросы, реклама) (извне).
* Решения по рекомендациям (от дирекции).

*Выходная информация:*

* Аналитические и финансовые отчеты (для дирекции)
* Рекомендации (для дирекции)
* Список поставщиков (для отдела поставок, склада)
* Список видов товаров (для отдела поставок)
* Отпускные цены (для торговых отделов).

1. **Отдел кадров**

Оформляет трудовые отношения с сотрудниками в соответствии с законодательством:

* Прием на работу.
* Перевод на другую должность.
* Увольнение.

*Входная информация:*

* Список торговых точек (от дирекции).
* Количество рабочих мест и требования к персоналу (от дирекции).
* Список сотрудников (от отдела кадров).
* Сведения о претендентах на работу в организации (извне).
* Заявления от сотрудников (из всех отделов).

Выходная информация:

* Список сотрудников (для экономического отдела и дирекции).
* Должностные обязанности сотрудников (для всех отделов).
* Программы внедрения сотрудника в рабочий процесс (для всех отделов).
* Контроль срока медицинской книжки (для всех отделов).
* Договора с заработной платой (для экономического отдела).

1. **Торговый отдел**

Занимается составлением заказов и всеми вопросами, связанными с товарами.

*Входная информация:*

* Заявка (от торговой точки).
* Отпускные цены (от экономического отдела).

*Выходная информация:*

* Заказ (для поставщиков, склада).
* Сведения о продажах (для экономического отдела).
* Сведения о списанных товарах (для экономического отдела).
* Учёт финансовых операций (для экономического отдела).

1. **Отдел поставок**

Ответственный за доставку и прием товара на склад, а также за качество доставленных товаров.

*Входная информация:*

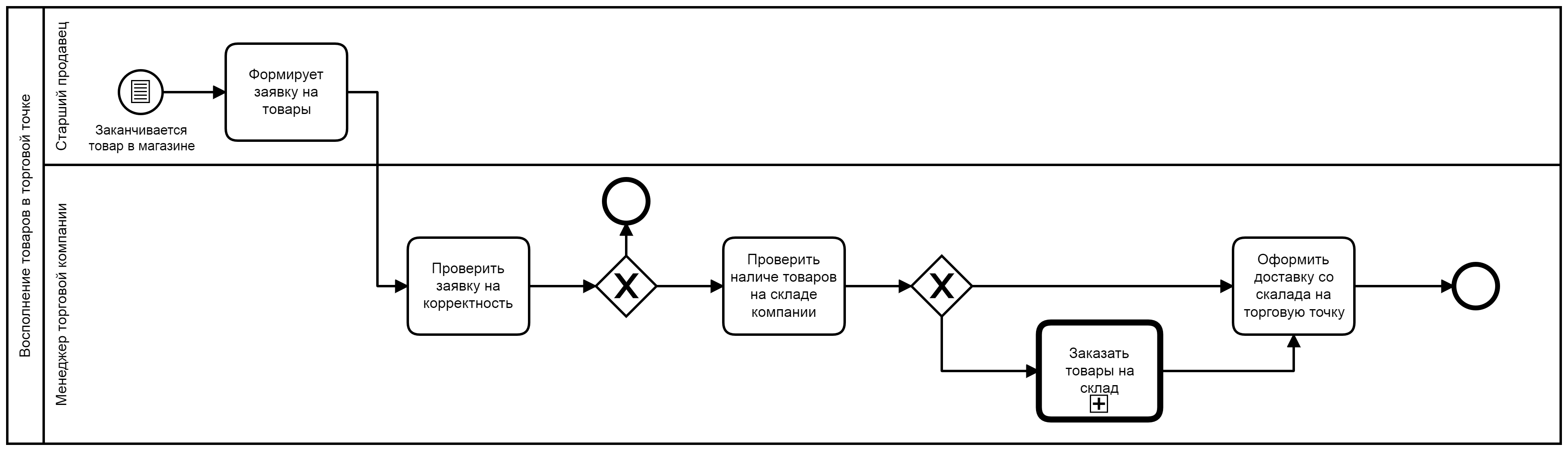
* Список поставщиков (от экономического отдела).
* Список видов товаров (от экономического отдела).
* Заказ (от торгового отдела).
* Заявка (от торговой точки).

*Выходная информация:*

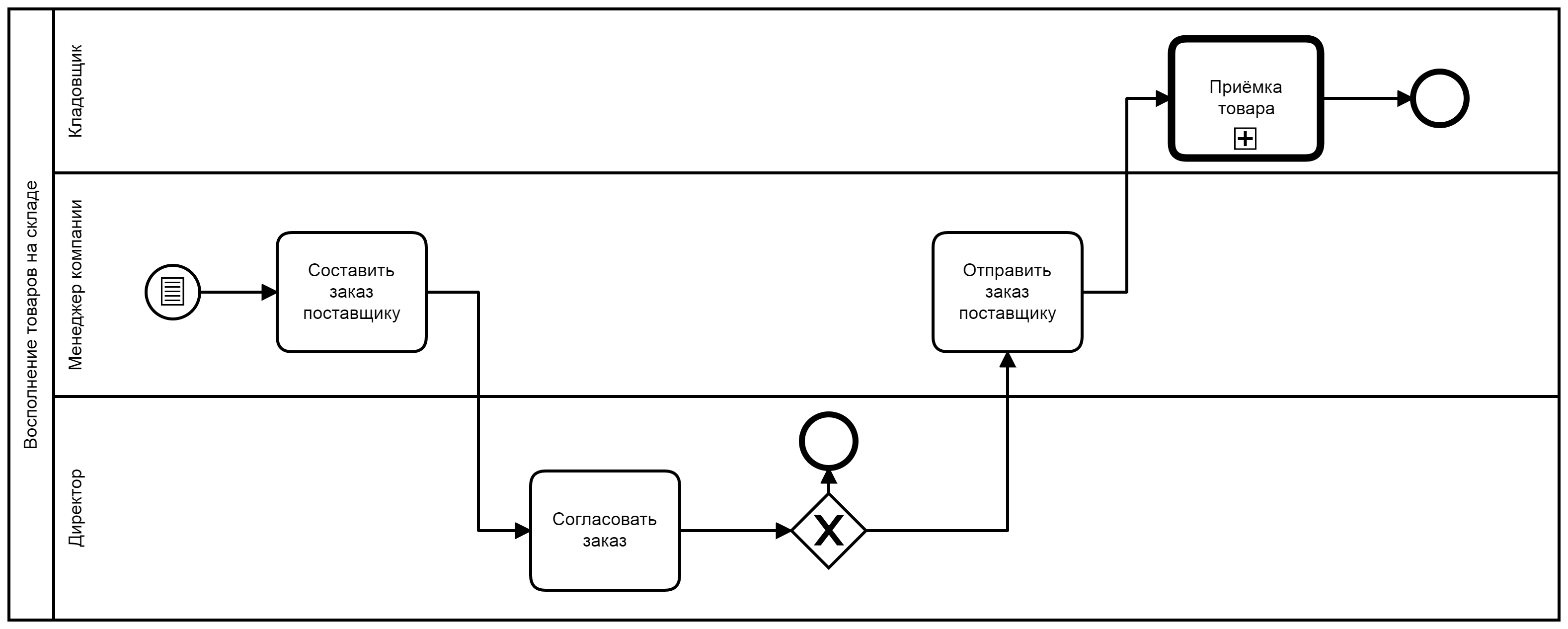
* Сведения о поставках (для экономического отдела).
* Акт о принятых и непринятых товаров на склад (я экономического отдела).

Бизнес-процессы торговой компании

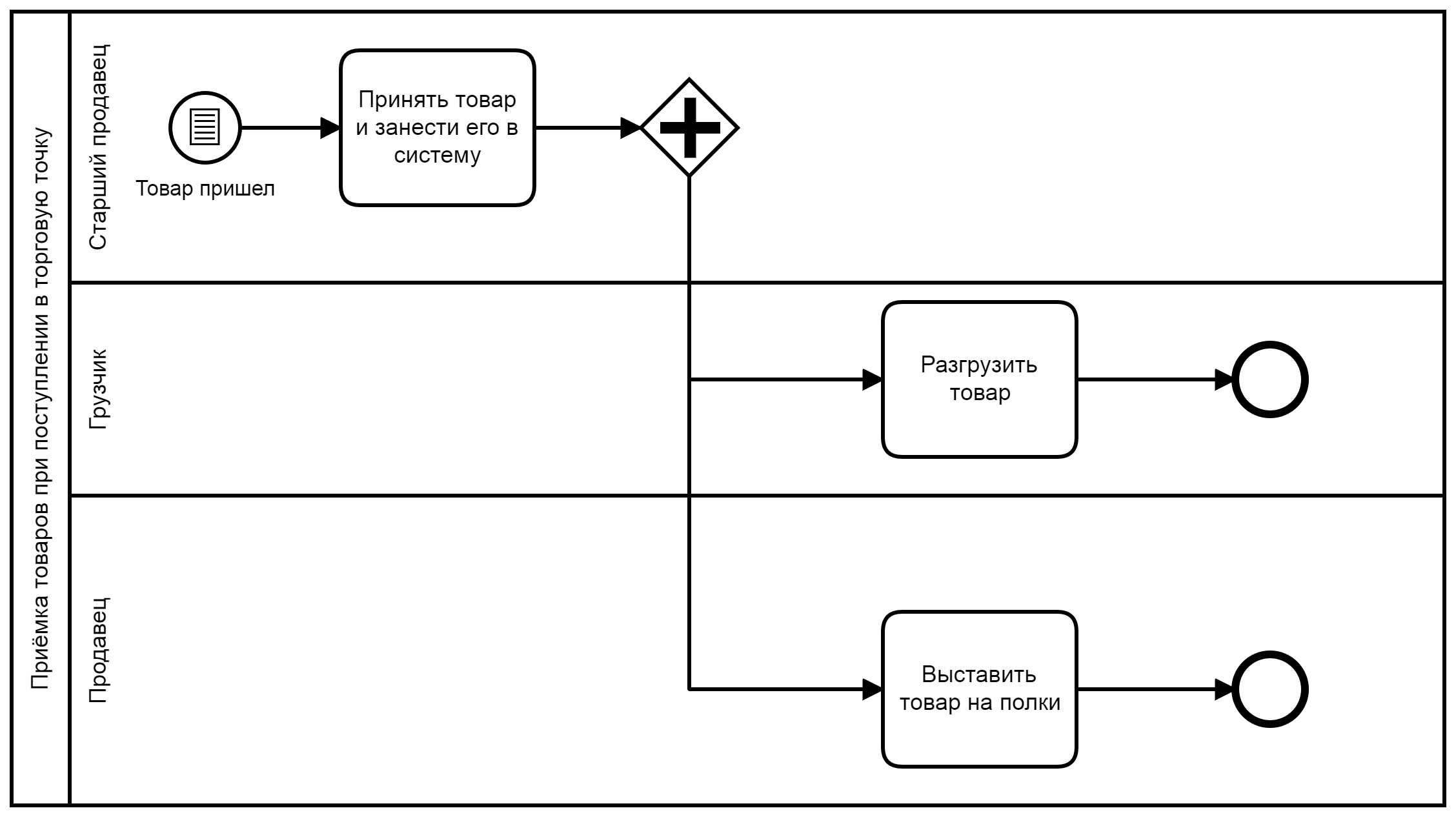
1. **Восполнение товаров в торговой точке**
   1. Старший продавец торговой точки составляет заявку на товары – фиксируются торговая точка, дата создания заявки, товары и заказанное количество товаров.
   2. Менеджер торговой компании обрабатывает заявку – организует доставку товара в торговую точку и оформляет заказы на товары при их отсутствии на складе. При поступлении товаров в торговую точку фиксируются торговая точка, товары, дата поступления товаров, количество единиц товаров и розничная цена.



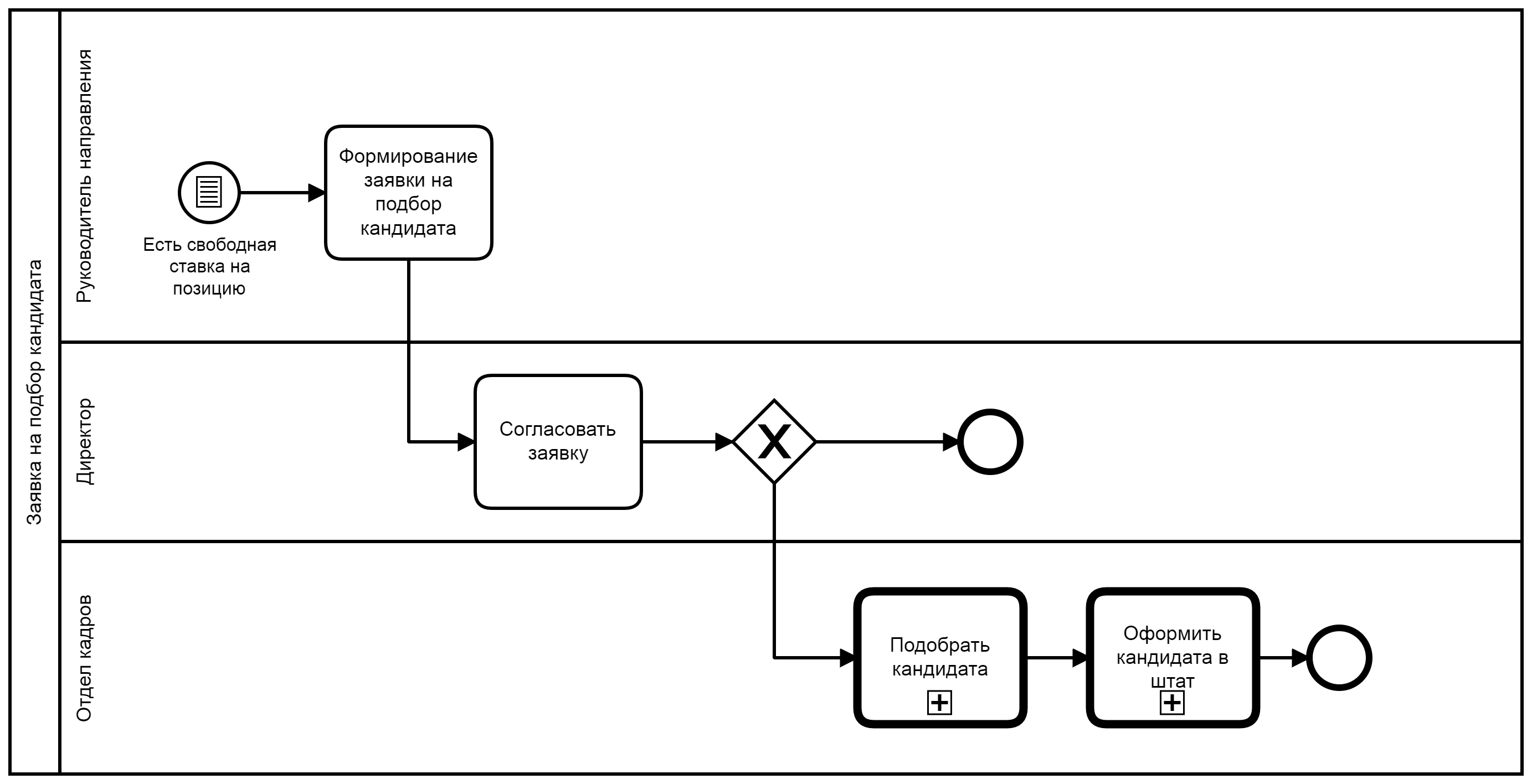
1. **Восполнение товаров на складе**
   1. Менеджер торговой компании составляет заказ на поставки товаров – фиксируются склад, дата создания заказа, поставки и заказанное количество товаров.
   2. Поставщик исполняет обязательства по заказу – организует доставку товара на склад. При поступлении товаров на склад фиксируются склад, поставки, дата выполнения поставки и количество единиц товаров.



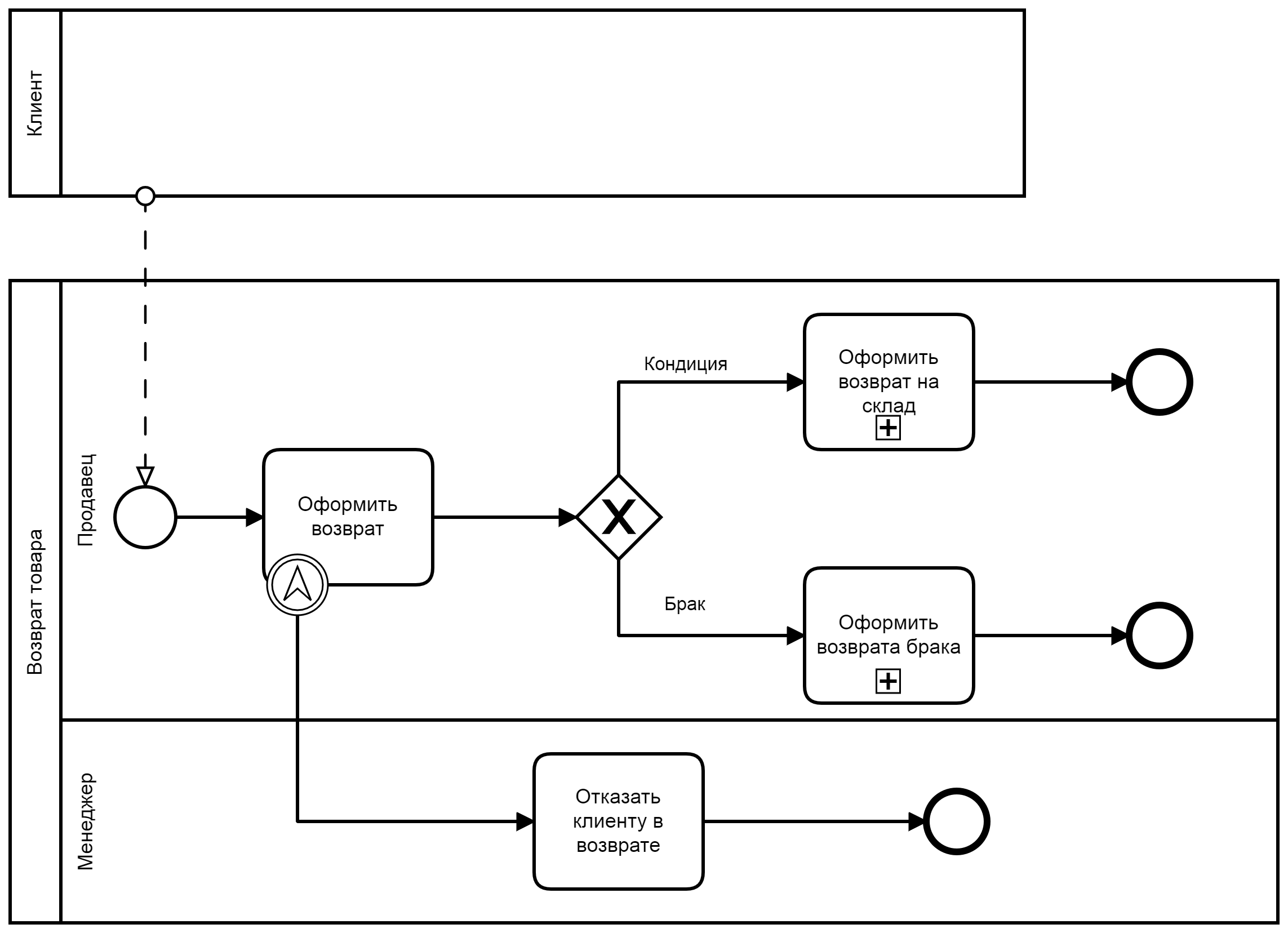
1. **Приемка товаров при поступлении в торговую точку**



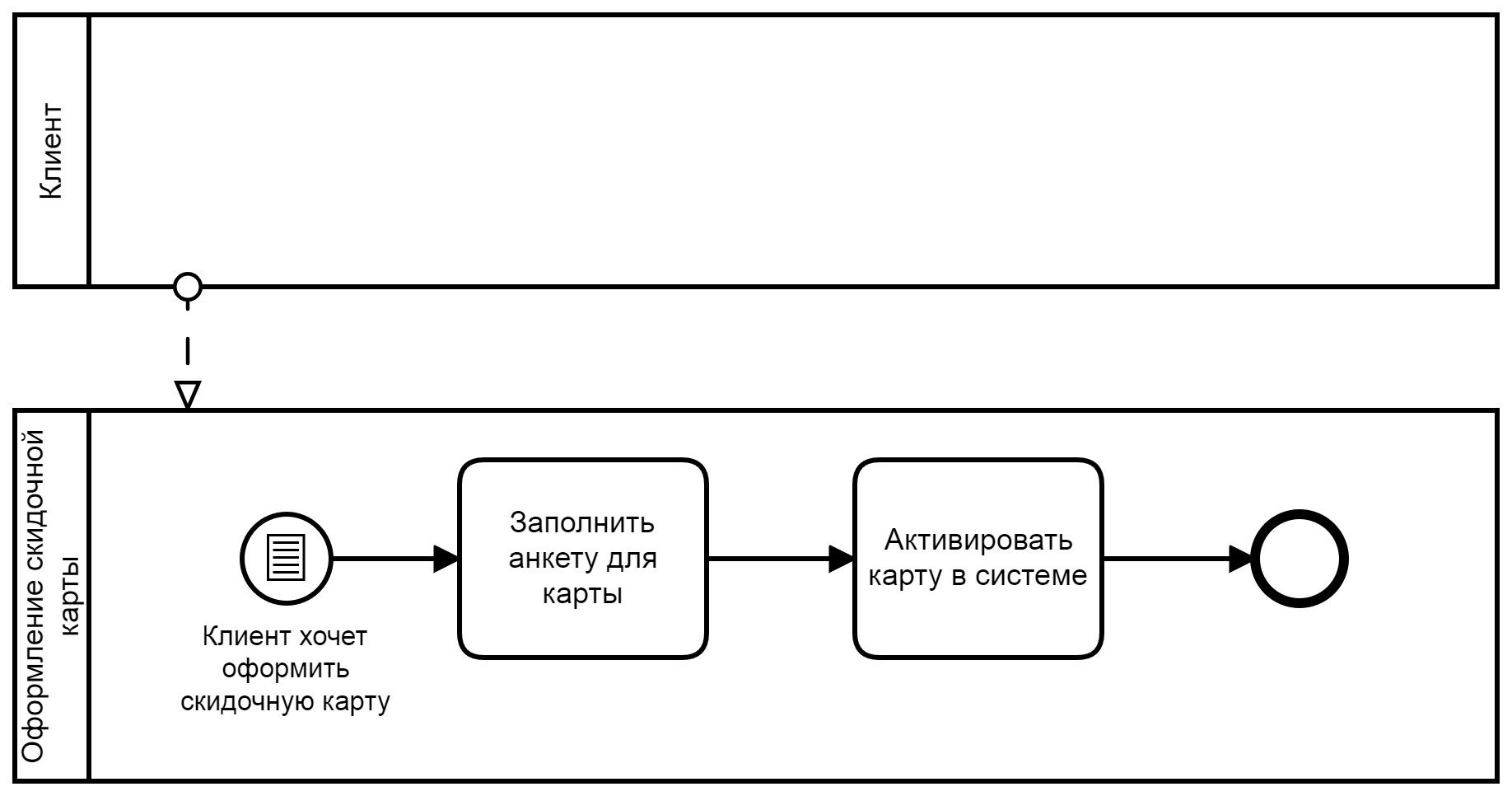
1. **Заявка на подбор кандидата**



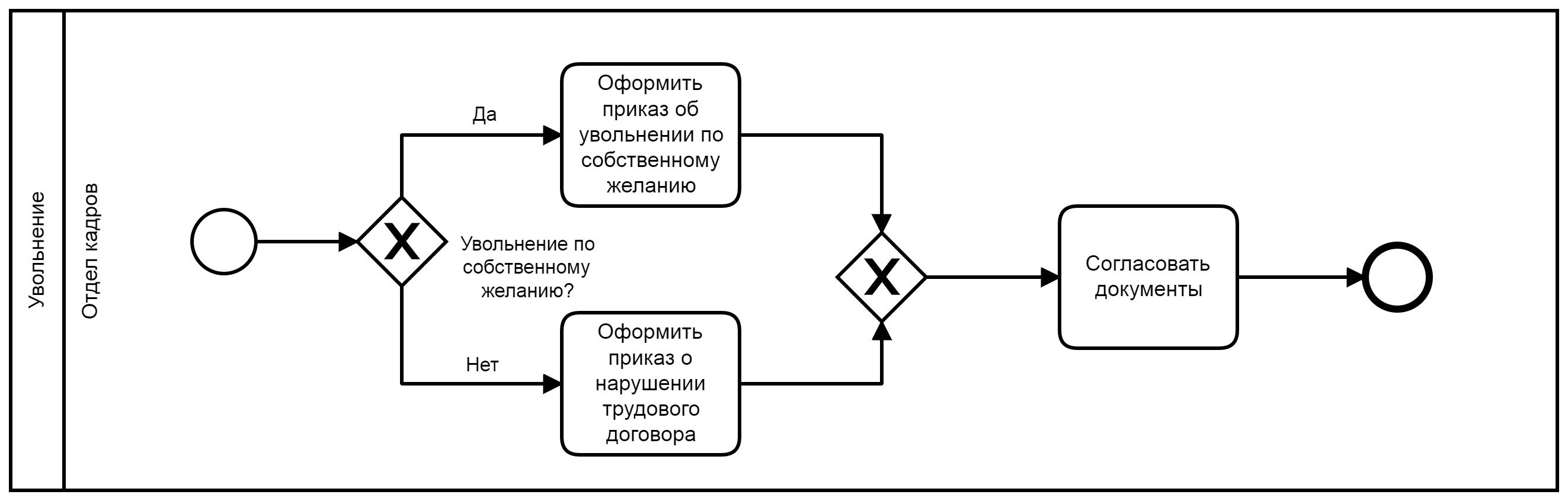
1. **Возврат товара**



1. **Оформление скидочной карты**



1. **Увольнение**



1. Инфологическое проектирование

**Сущность** – это реальный или представляемый объект предметной области, информация о котором должна сохраняться и быть доступна.

**Атрибут** – поименованная характеристика сущности, определяющая его свойства и принимающая значения из некоторого множества. Каждому атрибуту дается имя, уникальное в пределах сущности («Имя сущности».«Имя атрибута»).

В нашей предметной области стержневыми являются сущности: поставщик, товар, склад, сотрудник торговой компании, торговая точка, сотрудник, заказ и заявка.

Один поставщик может поставлять множество различных товаров, в то же время, один товар может поставляться разными поставщиками. Для формализации связи "многие ко многим" вводится ассоциативная сущность "поставка".

Для учёта товаров, поступающих на склад (склады) компании, вводится дополнительная ассоциативная сущность "распределение". Сущность "распределение" связана с сущностями "поставщик" и "товар" через сущность "поставка".

Каждый товар относится к определённому типу. Чтобы база данных имела 3 нормальную форму, вводится дополнительная обозначающая сущность "тип товара", которая хранит информацию о типе товара и его описание.

Заказ, отправляемый поставщику, может включать в себя несколько различных товаров. Чтобы база данных имела 3 нормальную форму, вводится дополнительная характеристическая сущность "позиция заказа". Аналогично, заявка, поступающая сотруднику торговой компании (менеджеру), может состоять из нескольких товаров. Чтобы база данных имела 3 нормальную форму, вводится дополнительная характеристическая сущность "позиция заявки".

Для учета товаров, приходящих со склада на торговые точки, вводится дополнительная ассоциативная сущность "поступление", которая связывает сущности "товар", "торговая точка" и "цена". Сущность "цена" содержит информацию о цене и сроке действия этой цены. Один и тот же товар в разных торговых точках может продаваться по разной цене, для учета этого условия и сохранения 3 нормальной формы базы данных, вводится ассоциативная сущность "цена".

Торговые точки относятся к определённому типу. Чтобы обозначить тип торговой точки и не нарушать 3 нормальную форму, вводится дополнительная обозначающая сущность "тип торговой точки".

Для фиксации продажи товара вводится ассоциативная сущность "реализация", которая связывает сущности "клиент", "товар", "торговая точка" и "сотрудник".

Каждая торговая точка не только приносит доход, но и имеет накладные расходы, такие как аренда, оплата коммунальных услуг и др. Чтобы база данных имела 3 нормальную форму, информация об этих платежах вынесена в отдельную характеристическую сущность "платеж", которая показывает, какие накладные расходы имеет каждая торговая точка. Также фиксируются дата и время совершения платежа. Чтобы не нарушать 3 нормальную форму, вводится обозначающая сущность "тип платежа".

Сотрудники занимают определенные должности, для фиксации должности сотрудника и сохранения 3 нормальной формы базы данных, вводится обозначающая сущность "должность".

Список сущностей предметной области и их атрибуты

1. **Поставщик** – стержневая сущность

* \*ID – указывающий
* Имя поставщика - указывающий
* Дата начала сотрудничества - описательный
* Дата окончания сотрудничества - описательный
* Причина прекращения сотрудничества - описательный
* Банковские реквизиты - описательный
* Контактная информация - описательный
* Адрес - описательный

1. **Поставка** – ассоциативная сущность

* \*ID - указывающий
* \*ID поставщика - вспомогательный
* \*Штрих-код товара - вспомогательный
* Оптовая цена - описательный
* Максимальный объем поставки - описательный
* Минимальный объем поставки - описательный
* Дата производства товара – описательный

1. **Товар** – стержневая сущность

* \*Штрих-код - указывающий
* \*ID типа товара - вспомогательный
* Наименование - вспомогательный
* Описание товара - описательный
* Срок хранения - описательный

1. **Тип товара** – обозначающая сущность

* \*ID - указывающий
* Название типа товара - указывающий
* Характеристика типа товара - описательный

1. **Позиция заказа** – характеристическая сущность

* \*ID - указывающий
* \*ID заказа - вспомогательный
* \*ID поставки - вспомогательный
* Количество единиц товара - описательный

1. **Заказ** – стержневая сущность

* \*ID - указывающий
* \*ID сотрудника - вспомогательный
* Дата создания заказа - описательный

1. **Заявка** – стержневая сущность

* \*ID – указывающий - вспомогательный
* \*ID сотрудника - вспомогательный
* Статус обработки - описательный
* Дата создания - описательный
* Дата последнего изменения статуса - описательный

1. **Позиция заявки** – характеристическая сущность

* \*ID - указывающий
* \*ID заявки - вспомогательный
* \*Штрих-код товара - вспомогательный
* Количество единиц товара - описательный

1. **Клиент** – стержневая сущность

* \*ID - указывающий
* Номер карточки - указывающий
* ФИО - указывающий
* Дата рождения - указывающий
* Дата выдачи карточки - описательный
* Дата активации карточки - описательный
* Срок действия карточки - описательный

1. **Сотрудник** – стержневая сущность

* \*ID - указывающий
* \*ID места работы - вспомогательный
* \*ID должности - вспомогательный
* ФИО - указывающий
* Дата рождения - указывающий
* Зарплата - описательный
* Дата вступления в должность - описательный
* Дата ухода с должности - описательный

1. **Поступление** – ассоциативная сущность

* \*ID - указывающий
* \*ID торговой точки - вспомогательный
* \*Штрих-код товара – вспомогательный
* \*ID цены - вспомогательный
* Дата поступления товара - описательный
* Количество - описательный

1. **Должность** – обозначающая сущность

* \*ID - указывающий
* Название - указывающий
* Уровень доступа - описательный
* Зарплата минимальная - описательный
* Зарплата максимальная - описательный

1. **Торговая точка** – стержневая сущность

* \*ID - указывающий
* \*ID типа торговой точки – вспомогательный
* Адрес - описательный
* Дата открытия - описательный
* Дата закрытия – описательный

1. **Тип торговой точки** – обозначающая сущность

* \*ID - указывающий
* Название типа - указывающий
* Характеристика типа - описательный

1. **Реализация** – ассоциативная сущность

* \*ID - указывающий
* \*ID торговой точки - вспомогательный
* \*Штрих-код товара - вспомогательный
* \*ID сотрудника торговой точки - вспомогательный
* \*ID клиента - вспомогательный
* Цена - описательный
* Количество проданного товара - описательный
* Дата и время продажи - описательный

1. **Цена** – ассоциативная сущность

* \*ID - указывающий
* Цена - описательный
* Дата начала действия цены - описательный
* Дата окончания действия цены - описательный

1. **Распределение** – ассоциативная сущность

* \*ID – указывающий
* \*ID поставки - вспомогательный
* \*ID склада - вспомогательный
* Дата выполнения поставки
* Количество единиц товара

1. **Платеж** – характеристическая сущность

* \*ID - указывающий
* \*ID торговой точки – вспомогательный
* \*ID типа платежа – вспомогательный
* Дата и время оплаты
* Сумма

1. **Тип** **платежа** – обозначающая сущность

* \*ID - указывающий
* Наименование платежа – указывающий
* Характеристика типа платежа

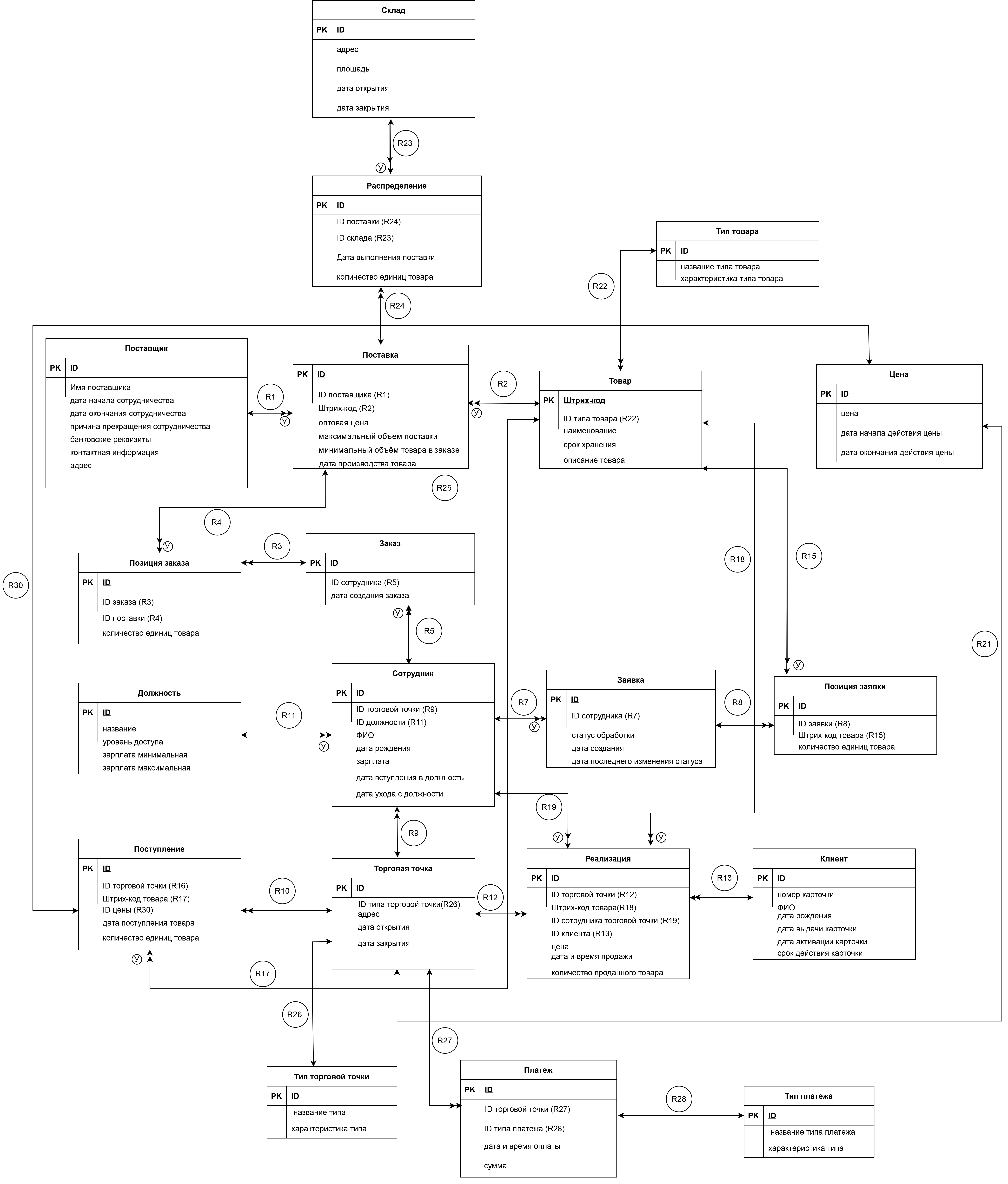
1. **Склад** – стержневая сущность

* Адрес
* Площадь
* Дата открытия
* Дата закрытия

Множественность и условность связей между сущностями определены в таблице:

|  |  |  |  |
| --- | --- | --- | --- |
| **Первая сущность** | **Вторая сущность** | **Множественность и условность связи** | **Пояснение** |
| Поставщик | Поставка | 1:Му | Один поставщик может поставлять разные товары. Также может быть поставщик, не поставляющий в данный момент никаких товаров |
| Склад | Распределение | 1:Му | Распределение товара, поступающего от поставщика, происходит между одним или несколькими складами  Может существовать склад, на который товар еще не поступал. |
| Поставка | Распределение | 1:М | Товары из поставки распределяются по складам. Одна поставка может быть распределена по нескольким складам. |
| Поставка | Позиция заказа | 1:Му | Товары, поставляемые некоторым поставщиком, могут составлять разные позиции заказов. Могут существовать поставки, еще не входившие ни в одну позицию заказа |
| Товар | Тип товара | 1:1 | Каждый товар относится к определенному виду |
| Товар | Поставка | 1:Му | Один товар может поставляться разными поставщиками. Также может существовать товар, который в данный момент не поставляется ни одним поставщиком. |
| Товар | Позиция заявки | 1:Му | Товар входит в позицию заявки. Один и тот же товар может составлять многие позиции заявок. Также может существовать товар, не вошедший ни в одну позицию заявки. |
| Товар | Поступление | 1:Му | Один и тот же товар может поступать несколько раз, а также может быть существовать товар, не поставленный ни разу. |
| Цена | Поступление | 1:Му | Поступая в торговую точку, товар имеет определенную цену для продажи. Один и тот же товар может иметь разную цену в разных торговых точках. |
| Товар | Реализация | 1:Му | Один и тот же товар может продаваться несколько раз, также может существовать товар, который еще ни разу не был продан |
| Заказ | Позиция заказа | 1:М | Заказ может состоять из одной или нескольких позиций заказа. |
| Сотрудник | Заказ | 1:Му | Один менеджер может формировать несколько заказов, в то время как заказ формируется конкретным менеджером. Может существовать менеджер, не сформировавший ни одного заказа. |
| Сотрудник | Заявка | 1:Му | Один менеджер может обрабатывать несколько заявок, в то время, как заявка обрабатывается конкретным менеджером. Может существовать менеджер, не обработавший еще ни одной заявки. |
| Должность | Сотрудник | 1:Му | Каждый сотрудник определенную должность. Может существовать свободная должность. |
| Заявка | Позиция заявки | 1:М | Заявка может состоять из одной или нескольких позиций. |
| Сотрудник | Заявка | 1:Му | Один сотрудник может формировать несколько заявок, в то время, как заявка формируется конкретным сотрудником. Может существовать сотрудник, не сформировавший еще ни одной заявки. |
| Должность | Сотрудник | 1: Му | Каждый сотрудник занимает определенную должность. Несколько разных сотрудников могут занимать одну и ту же должность.  Также может существовать должность, еще никем не занятая. |
| Сотрудник | Реализация | 1:Му | Реализация товара производится сотрудником (продавцом). Один продавец может реализовать множество товара. Также есть сотрудники, ни разу не реализовывавшие товар. |
| Торговая точка | Тип торговой точки | 1:1 | Каждая торговая точка относится к определённому типу. |
| Торговая точка | Сотрудник | 1:М | Сотрудники работают в определенной торговой точке. В одной торговой точке может работать несколько сотрудников. |
| Торговая точка | Поступление | 1:М | В торговую точку поступает товар, он может поступать несколько раз в разное время. |
| Торговая точка | Цена | 1:М | Для каждой торговой точки существует цена на определенный товар, которая может отличаться в разных точках. |
| Торговая точка | Реализация | 1:М | Торговая точка реализует разный товар, при этом один и тот же товар может реализовываться в разных точках. |
| Торговая точка | Платеж | 1:М | Каждая торговая точка осуществляет платежи (аренда, коммунальные расходы, выплаты сотрудникам), которых может быть несколько |
| Клиент | Реализация | 1:М | Реализация составляется из покупок клиентов. Один клиент может купить множество товаров, в то время, как единица товара покупается конкретным клиентом. |
| Реализация | Цена | 1:1 | Реализация товара происходит по конкретной актуальной цене. |
| Платеж | Тип платежа | 1:1 | Каждый платеж относится к конкретному типу. |

ER-диаграмма



1. Логическое проектирование

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Сущность (название таблицы в БД) | Атрибут | Атрибут в БД | Тип данных | Not Null | Первичный ключ | Внешний ключ | Ограничение |
| Поставщик (supplier) | ID | id | integer |  |  |  | Генерация из последовательности |
| имя поставщика | name | character(30) |  |  |  |  |
| дата начала сотрудничества | begin\_date | timestamptz |  |  |  |  |
| дата окончания сотрудничества | end\_date | timestamptz |  |  |  | Больше, чем дата начала сотрудничества |
| причина прекращения сотрудничества | reason\_end | character(280) |  |  |  |  |
| банковские реквизиты | bank\_details | character(280) |  |  |  |  |
| контактная информация | contacts | character(280) |  |  |  |  |
| адрес | address | character(280) |  |  |  |  |
| Поставка (supply) | ID | id | integer |  |  |  | Генерация из последовательности |
| ID поставщика | provider\_id | integer |  |  |  |  |
| Штрих-код товара | product\_barcode | character(13) |  |  |  |  |
| оптовая цена | trade\_price | money |  |  |  | Больше 0 |
| максимальный объем поставки | max\_size | integer |  |  |  | Больше 0; больше, чем минимальный объем поставки |
| минимальный объем поставки | min\_size | integer |  |  |  | Больше 0 |
| дата производства товара | production\_date | timestamptz |  |  |  |  |
| Товар (product) | Штрих-код | barcode | character(13) |  |  |  |  |
| ID типа товара | product\_type\_id | integer |  |  |  |  |
| наименование | name | character(50) |  |  |  |  |
| описание товара | description | character(280) |  |  |  |  |
| срок хранения | shelf\_life | interval |  |  |  |  |
| Тип товара (product\_type) | ID | id | integer |  |  |  | Генерация из последовательности |
| название типа товара | name | character(30) |  |  |  | Уникальный |
| характеристика типа товара | specification | character(280) |  |  |  |  |
| Позиция заказа (order\_item) | ID | id | integer |  |  |  | Генерация из последовательности |
| ID заказа | order\_id | integer |  |  |  |  |
| ID поставки | supply\_id | integer |  |  |  |  |
| количество единиц товара | amount | integer |  |  |  | Больше 0 |
| Заказ (orders) | ID | id | integer |  |  |  |  |
| ID сотрудника | employee\_id | integer |  |  |  |  |
| дата создания заказа | creation\_date | timestamptz |  |  |  |  |
| Заявка (application) | ID | id | integer |  |  |  | Генерация из последовательности |
| ID сотрудника | employee\_id | integer |  |  |  |  |
| статус обработки | status | character(20) |  |  |  |  |
| дата создания | creation\_date | timestamptz |  |  |  |  |
| дата последнего изменения статуса | changing\_status\_date | timestamptz |  |  |  |  |
| Позиция заявки (application\_item) | ID | id | integer |  |  |  | Генерация из последовательности |
| ID заявки | application\_id | integer |  |  |  |  |
| Штрих-код товара | product\_barcode | character(13) |  |  |  |  |
| количество единиц товара | amount | integer |  |  |  | Больше 0 |
| Клиент (client) | ID | id | integer |  |  |  | Генерация из последовательности |
| номер карточки | card\_number | character(20) |  |  |  | Уникальный |
| ФИО | fullname | character(50) |  |  |  |  |
| дата рождения | birth\_date | timestamptz |  |  |  |  |
| дата выдачи карточки | creation\_card\_date | timestamptz |  |  |  |  |
| дата активации карточки | activation\_card\_date | timestamptz |  |  |  | Больше, чем дата выдачи карточки |
| срок действия карточки | card\_shelf\_life | interval |  |  |  |  |
| Сотрудник (employee) | ID | id | integer |  |  |  | Генерация из последовательности |
| ID торговой точки | outlet\_id | integer |  |  |  |  |
| ID должности | position\_id | integer |  |  |  |  |
| ФИО | fullname | character(50) |  |  |  |  |
| дата рождения | birth\_date | timestamptz |  |  |  |  |
| зарплата | salary | money |  |  |  | Больше 0 |
| дата вступления в должность | date\_begin | timestamptz |  |  |  |  |
| дата ухода с должности | date\_end | timestamptz |  |  |  | Больше, чем дата вступления в должность |
| Поступление (coming) | ID | id | integer |  |  |  | Генерация из последовательности |
| ID торговой точки | outlet\_id | integer |  |  |  |  |
| Штрих-код товара | product\_barcode | character(13) |  |  |  |  |
| ID цены | price\_id | integer |  |  |  |  |
| дата поступления товара | coming\_date | timestamptz |  |  |  |  |
| количество | amount | integer |  |  |  | Больше или равно 0 |
| Должность (position) | ID | id | integer |  |  |  | Генерация из последовательности |
| название | name | character(50) |  |  |  |  |
| уровень доступа | access\_level | integer |  |  |  | Больше или равно 0 |
| зарплата минимальная | min\_salary | money |  |  |  | Больше 0 |
| зарплата максимальная | max\_salary | money |  |  |  | Больше 0 |
| Торговая точка (outlet) | ID | id | integer |  |  |  | Генерация из последовательности |
| ID типа торговой точки | outlet\_type\_id | integer |  |  |  |  |
| адрес | address | character(280) |  |  |  |  |
| дата открытия | open\_date | timestamptz |  |  |  |  |
| дата закрытия | close\_date | timestamptz |  |  |  | Больше, чем дата открытия |
| Тип торговой точки (outlet\_type) | ID | id | integer |  |  |  | Генерация из последовательности |
| название типа | name | character(30) |  |  |  |  |
| характеристика типа | specification | character(280) |  |  |  |  |
| Реализация (selling) | ID | id | integer |  |  |  | Генерация из последовательности |
| ID торговой точки | outlet\_id | integer |  |  |  |  |
| Штрих-код товара | product\_barcode | character(13) |  |  |  |  |
| ID сотрудника | employee | integer |  |  |  |  |
| ID клиента | client\_id | integer |  |  |  |  |
| цена | price | money |  |  |  | Больше 0 |
| количество проданного товара | amount | integer |  |  |  | Больше 0 |
| дата и время продажи | sale\_datetime | timestamptz |  |  |  |  |
| Цена (price) | ID | id | integer |  |  |  | Генерация из последовательности |
| цена | item\_price | money |  |  |  | Больше 0 |
| дата начала действия цены | begin\_date | timestamptz |  |  |  |  |
| дата окончания действия цены | end\_date | timestamptz |  |  |  | Больше, чем дата начала действия цены |
| Распределение (distribution) | ID | id | integer |  |  |  | Генерация из последовательности |
| ID поставки | supply\_id | integer |  |  |  |  |
| ID склада | warehouse\_id | integer |  |  |  |  |
| дата выполнения поставки | delivery\_date | timestamptz |  |  |  |  |
| количество единиц товара | amount | integer |  |  |  | Больше 0 |
| Платеж (payment) | ID | id | integer |  |  |  | Генерация из последовательности |
| ID торговой точки | outlet\_id | integer |  |  |  |  |
| ID типа платежа | payment\_id | integer |  |  |  |  |
| дата и время оплаты | datetime | timestamptz |  |  |  |  |
| сумма | amount | money |  |  |  | Больше 0 |
| Тип платежа (payment\_type) | ID | id | integer |  |  |  | Генерация из последовательности |
| наименование платежа | name | character(30) |  |  |  |  |
| характеристика типа платежа | specification | character(50) |  |  |  |  |
| Склад (warehouse) | ID | id | integer |  |  |  | Генерация из последовательности |
| адрес | address | character(280) |  |  |  |  |
| площадь | area | integer |  |  |  | Больше 0 |
| дата открытия | open\_date | timestamptz |  |  |  |  |
| дата закрытия | close\_date | timestamptz |  |  |  | Больше, чем дата открытия |

Скрипт для создания базы данных

create table if not exists supplier  
(  
 id int generated always as identity primary key,  
 name character(30) not null,  
 begin\_date timestamptz,  
 end\_date timestamptz,  
 reason\_end character(280),  
 bank\_details character(280) not null,  
 contacts character(280) not null,  
 address character(280) not null,  
 constraint end\_date\_check check ( supplier.end\_date > supplier.begin\_date )  
);  
  
create table if not exists product\_type  
(  
 id int generated always as identity primary key,  
 name character(30) not null unique,  
 specification character(290) not null  
);  
  
create table if not exists product  
(  
 barcode character(13) primary key,  
 product\_type\_id int references product\_type,  
 name character(50) not null,  
 description character(280) not null,  
 shelf\_life interval not null  
);  
  
create table if not exists supply  
(  
 id int generated always as identity primary key,  
 supplier\_id int references supplier,  
 product\_barcode character(13) references product,  
 trade\_price money not null,  
 min\_size int not null,  
 max\_size int not null,  
 production\_date timestamptz not null,  
 constraint trade\_price\_positive check ( supply.trade\_price > 0::money ),  
 constraint max\_size\_more\_min\_size check ( supply.max\_size > supply.min\_size )  
);  
  
create table if not exists outlet\_type  
(  
 id int generated always as identity primary key,  
 name character(30) not null unique,  
 specification character(290) not null  
);  
  
create table if not exists outlet  
(  
 id int generated always as identity primary key,  
 outlet\_type\_id int references outlet\_type,  
 address character(280) not null unique,  
 open\_date timestamptz not null,  
 close\_date timestamptz,  
 constraint open\_date\_less\_close\_date check (  
 outlet.open\_date < outlet.close\_date  
 )  
);  
  
create table if not exists position  
(  
 id int generated always as identity primary key,  
 name character(50) not null unique,  
 access\_level int not null,  
 min\_salary money not null,  
 max\_salary money not null,  
 constraint nim\_salary\_less\_then\_max\_salary check (  
 position.min\_salary > position.max\_salary  
 ),  
 constraint access\_level\_non\_negative check (  
 position.access\_level >= 0  
 ),  
 constraint min\_salary\_non\_negative check (  
 position.min\_salary >= 0::money  
 ),  
 constraint max\_salary\_non\_negative check (  
 position.max\_salary >= 0::money  
 )  
);  
  
create table if not exists employee  
(  
 id int generated always as identity primary key,  
 outlet\_id int references outlet,  
 position\_id int references position,  
 fullname character(100) not null,  
 personal\_data character(50) not null unique,  
 date timestamptz not null,  
 salary money not null,  
 begin\_date timestamptz not null,  
 end\_date timestamptz,  
 constraint fire\_date\_more\_begin\_date check (  
 employee.end\_date > employee.begin\_date  
 )  
);  
  
create table if not exists orders  
(  
 id int generated always as identity primary key,  
 employee\_id int references employee,  
 creation\_date timestamptz not null default now()  
);  
  
create table if not exists order\_item  
(  
 id int generated always as identity primary key,  
 order\_id int references orders,  
 supply\_id int references supply,  
 amount int not null,  
 constraint amount\_positive check ( order\_item.amount > 0 )  
);  
  
create table if not exists application  
(  
 id int generated always as identity primary key,  
 employee\_id int references employee,  
 status character(20) not null,  
 creation\_date timestamptz not null default now(),  
 changing\_status\_date timestamptz not null  
);  
  
create table if not exists application\_item  
(  
 id int generated always as identity primary key,  
 application\_id int references application,  
 product\_barcode character(13) references product,  
 amount int not null,  
 constraint unique (application\_id, product\_barcode, amount),  
 constraint amount\_is\_positive check ( application\_item.amount > 0 )  
);  
  
create table if not exists client  
(  
 id int generated always as identity primary key,  
 card\_number character(20) not null unique,  
 fullname character(100) not null,  
 birth\_date timestamptz not null,  
 creation\_card\_date timestamptz not null default now(),  
 activation\_card\_date timestamptz,  
 card\_shelf\_life interval not null,  
 constraint activation\_card\_date\_more\_than\_creation\_card\_date check (  
 client.activation\_card\_date > client.creation\_card\_date  
 )  
);  
  
create table if not exists price  
(  
 id int generated always as identity primary key,  
 price money not null,  
 begin\_date timestamptz not null,  
 end\_date timestamptz,  
 constraint price\_positive check (  
 price.price > 0::money  
 ),  
 constraint begin\_date\_more\_end\_date check (  
 price.begin\_date > price.end\_date  
 )  
);  
  
create table if not exists coming  
(  
 id int generated always as identity primary key,  
 outlet\_id int references outlet,  
 product\_barcode character(13) references product,  
 price\_id int references price,  
 coming\_date timestamptz not null,  
 amount int not null,  
 constraint amount\_non\_negative check ( amount >= 0 )  
);  
  
  
  
create table if not exists selling  
(  
 id int generated always as identity primary key,  
 outlet\_id int references outlet,  
 product\_barcode character(13) references product,  
 employee\_id int references employee,  
 client\_id int references client,  
 price money not null,  
 amount int not null,  
 sale\_datetime timestamptz not null,  
 constraint price\_positive check (  
 selling.price > 0::money  
 ),  
 constraint amount\_positive check (  
 selling.amount > 0  
 )  
);  
  
create table if not exists warehouse  
(  
 id int generated always as identity primary key,  
 address character(290) not null unique,  
 area int not null,  
 open\_date timestamptz not null,  
 close\_date timestamptz,  
 constraint area\_positive check (  
 warehouse.area > 0  
 ),  
 constraint open\_date\_more\_close\_date check (  
 warehouse.open\_date > warehouse.close\_date  
 )  
);  
  
create table if not exists distribution  
(  
 id int generated always as identity primary key,  
 supply\_id int references supply,  
 warehouse\_id int references warehouse,  
 delivery\_date timestamptz,  
 amount int,  
 constraint amount\_positive check (  
 distribution.amount > 0  
 )  
);  
  
create table if not exists payment\_type  
(  
 id int generated always as identity primary key,  
 name character(30) not null unique,  
 specification character(290) not null  
);  
  
create table if not exists payment  
(  
 id int generated always as identity primary key,  
 outlet\_id int references outlet,  
 payment\_type\_id int references payment\_type,  
 datetime timestamptz not null,  
 amount int,  
 constraint amount\_positive check (  
 payment.amount > 0  
 )  
);

1. Проектирование приложений

Программная реализация представляет из себя клиент-серверное приложение. Backend выполнен на языке программирования Java с использованием фреймворка Spring Boot и движком исполнения и автоматизации бизнес-процессов Camunda (приложенные схемы в разделе «Бизнес-процессы» являются исполняемыми). Frontend выполнен на языке программирования JavaScript с использованием библиотеки React. В качестве СУБД для информационной системы был использован PostgreSQL. Сервис развернут внутри Docker-контейнера.

Помимо базы данных, описанной в инфологическом проектировании, была использо-вана база данных H2 внутри Camunda для контроля за исполнением бизнес-процессов. Напрямую взаимодействие с этой базой данных не ведется, к базе обращается только Camunda.

Данные по историческим процессам пишутся в отдельную базу данных, чтобы избе-жать повышенной нагрузки при исполнении аналитических запросов.

Цели, задачи и требования

**Цель**

Обеспечение сотрудников компании информацией для выполнения их должностных обязанностей.

**Задачи**

* Обработка информации и выдача её в удобном для восприятия виде.
* Получение более эффективных способов решения управленческих задач за счёт внедрения математических методов и интеллектуальных систем.
* Производство достоверной, надёжной, своевременной и систематизированной информации.
* Освобождение работников от рутинной работы за счёт её автоматизации.
* Совершенствование структуры потоков информации и системы документооборота в фирме.

**Требования**

* Надежные средства защиты данных от несанкционированного доступа. Разграничение доступа к данным в соответствии с должностными обязанностями.
* Возможность удаленного доступа к функциям и данным системы.
* Автоматизация управления запасами в части оперативного получения информации об остатках на складе.
* Автоматизация управления закупками в части планирования закупок в разрезе поставщиков.
* Автоматизация контроля расчетов с поставщиками и клиентами.
* Получение управленческих отчетов в необходимых аналитических срезах — как детальных для менеджеров, так и агрегированных для руководителей подразделений и директоров компании.

Состав подсистем разрабатываемой ИС, назначение и функции каждого приложения ИС

**Информационное обеспечение**

Назначение подсистемы информационного обеспечения состоит в современном формировании и выдаче достоверной информации для принятия управленческих решений.

Информационное обеспечение — совокупность единой системы классификации и кодирования информации, унифицированных систем документации, схем информационных потоков, циркулирующих в организации, а также методология построения баз данных.

**Техническое обеспечение**

Техническое обеспечение — комплекс технических средств, предназначенных для работы информационной системы, а также соответствующая документация на эти средства и технологические процессы

Комплекс технических средств составляют:

* Компьютеры любых моделей.
* Устройства сбора, накопления, обработки, передачи и вывода информации.
* Устройства передачи данных и линий связи.
* Оргтехника и устройства автоматического съема информации.
* Эксплуатационные материалы.

Документацией оформляются предварительный выбор технических средств, организация их эксплуатации, технологический процесс обработки данных, технологическое оснащение. Документацию можно условно разделить на три группы:

* Общесистемную, включающую государственные и отраслевые стандарты по техническому обеспечению.
* Специализированную, содержащую комплекс методик по всем этапам разработки технического обеспечения.
* Нормативно-справочную, используемую при выполнении расчетов по техническому обеспечению.

**Математическое и программное обеспечение**

Математическое и программное обеспечение — совокупность математических методов, моделей, алгоритмов и программ для реализации целей и задач информационной системы, а также нормального функционирования комплекса технических средств.

К средствам математического обеспечения относятся:

* Средства моделирования процессов управления.
* Типовые задачи управления.
* Методы программирования, теории массового обслуживания.

В состав программного обеспечения входят общесистемные и специальные программные продукты, а также техническая документация.

К общесистемному программному обеспечению относятся комплексы программ, ориентированных на пользователей и предназначенных для решения типовых задач обработки информации. Они служат для расширения функциональных возможностей компьютеров, контроля и управления процессом обработки данных.

**Организационное обеспечение**

Организационное обеспечение — совокупность методов и средств, регламентирующих взаимодействие работников с техническими средствами и между собой в процессе разработки и эксплуатации информационной системы.

Организационное обеспечение реализует следующие функции:

* Анализ существующей системы управления организацией, где будет использоваться ИС, и выявление задач, подлежащих автоматизации.
* Подготовку задач к решению на компьютере, включая техническое задание на проектирование ИС и технико-экономическое обоснование ее эффективности.
* Разработку управленческих решений по составу и структуре организации, методологии решения задач, направленных на повышение эффективности системы управления.

**Правовое обеспечение**

Правовое обеспечение — совокупность правовых норм, определяющих создание, юридический статус и функционирование информационных систем, регламентирующих порядок получения, преобразования и использования информации.

Главной целью правового обеспечения является укрепление законности.

В состав правового обеспечения входят законы, указы, постановления государственных органов власти, приказы, инструкции и другие нормативные документы министерств, ведомств, организаций, местных органов власти. В правовом обеспечении можно выделить общую часть, регулирующую функционирование любой информационной системы, и локальную часть, регулирующую функционирование конкретной системы.

Правовое обеспечение этапов разработки информационной системы включает нормативные акты, связанные с договорными отношениями разработчика и заказчика и правовым регулированием отклонений от договора.

Правовое обеспечение этапов функционирования информационной системы включает:

* Статус информационной системы.
* Права, обязанности и ответственность персонала.
* Правовые положения отдельных видов процесса управления.
* Порядок создания и использования информации.

Категории пользователей, работающих с приложениями ИС, выполняемые ими функции, права пользователей, категории данных, с которыми они работают

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Обзор прав доступа | | Роль | | | | | | |
| Категория данных | Право доступа | Продавец | Старший продавец | Менеджер | Бухгалтер | Рекрутер | Директор | Администратор |
| Товары | Просмотр | 🗸 | 🗸 | 🗸 | 🗸 |  | 🗸 | 🗸 |
| Редактирование |  |  | 🗸 |  |  | 🗸 | 🗸 |
| Удаление |  |  |  |  |  | 🗸 | 🗸 |
| Поставщики | Просмотр |  |  | 🗸 | 🗸 |  | 🗸 | 🗸 |
| Редактирование |  |  | 🗸 |  |  | 🗸 | 🗸 |
| Удаление |  |  |  |  |  | 🗸 | 🗸 |
| Поставки | Просмотр |  |  | 🗸 | 🗸 |  | 🗸 | 🗸 |
| Редактирование |  |  | 🗸 |  |  | 🗸 | 🗸 |
| Удаление |  |  |  |  |  | 🗸 | 🗸 |
| Сотрудники | Просмотр |  |  |  | 🗸 | 🗸 | 🗸 | 🗸 |
| Редактирование |  |  |  |  | 🗸 | 🗸 | 🗸 |
| Удаление |  |  |  |  |  | 🗸 | 🗸 |
| Клиенты | Просмотр | 🗸 | 🗸 |  | 🗸 |  | 🗸 | 🗸 |
| Редактирование |  | 🗸 |  |  |  | 🗸 | 🗸 |
| Удаление |  |  |  |  |  | 🗸 | 🗸 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Обзор прав доступа | | Роль | | | | | | |
| Вид информации | Право доступа | Продавец | Старший продавец | Менеджер | Бухгалтер | Рекрутер | Директор | Администратор |
| Платежи | Просмотр |  |  |  | 🗸 |  | 🗸 | 🗸 |
| Редактирование |  |  |  | 🗸 |  | 🗸 | 🗸 |
| Удаление |  |  |  |  |  | 🗸 | 🗸 |
| Цены | Просмотр |  |  |  | 🗸 |  | 🗸 | 🗸 |
| Редактирование |  |  |  |  |  | 🗸 | 🗸 |
| Удаление |  |  |  |  |  | 🗸 | 🗸 |
| Заказы | Просмотр |  |  | 🗸 | 🗸 |  | 🗸 | 🗸 |
| Редактирование |  |  | 🗸 |  |  | 🗸 | 🗸 |
| Удаление |  |  |  |  |  | 🗸 | 🗸 |
| Заявки | Просмотр |  | 🗸 | 🗸 | 🗸 |  | 🗸 | 🗸 |
| Редактирование |  | 🗸 |  |  |  | 🗸 | 🗸 |
| Удаление |  |  |  |  |  | 🗸 | 🗸 |
| Склады | Просмотр |  |  | 🗸 | 🗸 |  | 🗸 | 🗸 |
| Редактирование |  |  | 🗸 |  |  | 🗸 | 🗸 |
| Удаление |  |  |  |  |  | 🗸 | 🗸 |
| Торговые точки | Просмотр |  | 🗸 |  | 🗸 |  | 🗸 | 🗸 |
| Редактирование |  | 🗸 |  |  |  | 🗸 | 🗸 |
| Удаление |  |  |  |  |  | 🗸 | 🗸 |

Отчеты

Формирование отчета представляет собой исполнение sql-запроса на сервере и отображение пользователю в приложении. Пользователь может скачать результат выполнения запроса в форматах .csv и .xls. В качестве протокола взаимодействия используется HTTP. Вызов процесса формирование отчета доступен по нажатию соответствующей кнопки в приложении с указанием определенных параметров либо с помощью прямого обращения к серверу с помощью http-запроса с передачей параметров в теле запроса или в строке поиска.

Запросы для формирования отчетов

1. **Получить перечень и общее число поставщиков указанного вида товара либо некоторого товара в объеме не менее заданного за весь период сотрудничества либо за указанный период**

-- перечень поставщиков указанного вида товара

-- в объеме не менее заданного за весь период

SELECT DISTINCT supplier\_id

FROM supply

WHERE max\_size > ?

AND product\_barcode IN (SELECT barcode

FROM product

WHERE product\_type\_id = ?

);

-- перечень поставщиков указанного вида товара

-- в объеме не менее заданного за указанный период

SELECT DISTINCT supplier\_id

FROM supply

WHERE max\_size > ?

AND production\_date >= ?

AND production\_date <= ?

AND product\_barcode IN (SELECT barcode

FROM product

WHERE product\_type\_id = ?

);

-- перечень поставщиков некоторого товара

-- в объеме не менее заданного за весь период

SELECT DISTINCT supplier\_id

FROM supply

WHERE max\_size > ?

AND product\_barcode = ?;

-- перечень поставщиков некоторого товара

-- в объеме не менее заданного за указанный период

SELECT DISTINCT supplier\_id

FROM supply

WHERE max\_size > ?

AND production\_date >= ?

AND production\_date <= ?

AND product\_barcode = ?;

-- общее число поставщиков указанного вида товара

-- в объеме не менее заданного за весь период

SELECT COUNT(DISTINCT supplier\_id)

FROM supply

WHERE max\_size > ?

AND product\_barcode IN (SELECT barcode

FROM product

WHERE product\_type\_id = ?

);

-- общее число поставщиков указанного вида товара

-- в объеме не менее заданного за указанный период

SELECT COUNT(DISTINCT supplier\_id)

FROM supply

WHERE max\_size > ?

AND production\_date >= ?

AND production\_date <= ?

AND product\_barcode IN (SELECT barcode

FROM product

WHERE product\_type\_id = ?

);

-- общее число поставщиков некоторого товара

-- в объеме не менее заданного за весь период

SELECT COUNT(DISTINCT supplier\_id)

FROM supply

WHERE max\_size > ?

AND product\_barcode = ?;

-- общее число поставщиков некоторого товара

-- в объеме не менее заданного за указанный период

SELECT COUNT(DISTINCT supplier\_id)

FROM supply

WHERE max\_size > ?

AND production\_date >= ?

AND production\_date <= ?

AND product\_barcode = ?;

1. **Получить перечень и общее число покупателей указанного вида товара за некоторый период либо сделавших покупку товара в объеме не менее заданного**

-- перечень покупателей указанного вида товара за некоторый период

SELECT DISTINCT client\_id

FROM selling

WHERE sale\_datetime >= ?

AND sale\_datetime <= ?

AND product\_barcode IN (SELECT barcode

FROM product

WHERE product\_type\_id = ?

);

-- общее число покупателей указанного вида товара за некоторый период

SELECT COUNT(DISTINCT client\_id)

FROM selling

WHERE sale\_datetime >= ?

AND sale\_datetime <= ?

AND product\_barcode IN (SELECT barcode

FROM product

WHERE product\_type\_id = ?

);

-- перечень покупателей указанного вида товара,

-- сделавших покупку товара в объеме не менее заданного

SELECT DISTINCT client\_id

FROM selling

WHERE amount >= ?

AND product\_barcode IN (SELECT barcode

FROM product

WHERE product\_type\_id = ?

);

-- общее число покупателей указанного вида товара,

-- сделавших покупку товара в объеме не менее заданного

SELECT COUNT(DISTINCT client\_id)

FROM selling

WHERE amount >= ?

AND product\_barcode IN (SELECT barcode

FROM product

WHERE product\_type\_id = ?

);

1. **Получить номенклатуру и объем товаров в указанной торговой точке**

-- номенклатура и объем товаров в указанной торговой точке

SELECT p.\*, ca.coming\_amount - sa.selling\_amount AS amount

FROM (SELECT product\_barcode, SUM(amount) AS coming\_amount

FROM coming

WHERE outlet\_id = ?

GROUP BY product\_barcode

) AS ca

LEFT JOIN (SELECT product\_barcode, SUM(amount) AS selling\_amount

FROM selling

WHERE outlet\_id = ?

GROUP BY product\_barcode

) AS sa

ON ca.product\_barcode = sa.product\_barcode

INNER JOIN product AS p

ON ca.product\_barcode = p.barcode;

1. **Получить актуальные сведения об объеме и ценах на указанный товар по всем торговым точкам, по торговым точкам заданного типа, по конкретной торговой точке**

-- актуальные сведения об объеме и ценах на указанный товар

-- по всем торговым точкам

SELECT ca.product\_id, ca.coming\_amount - sa.selling\_amount AS amount, pr.item\_price

FROM (SELECT outlet\_id, product\_id, SUM(amount) AS coming\_amount

FROM coming

WHERE product\_id = ?

GROUP BY outlet\_id

) AS ca

LEFT JOIN (SELECT outlet\_id, product\_id, SUM(amount) AS selling\_amount

FROM selling

WHERE product\_id = ?

GROUP BY outlet\_id

) AS sa

ON ca.product\_id = sa.product\_id

AND ca.outlet\_id = sa.outlet\_id

INNER JOIN (

SELECT p.outlet\_id, p.product\_id, r.item\_price

FROM (SELECT id, item\_price

FROM price

WHERE relevance = TRUE

) AS r

INNER JOIN (SELECT outlet\_id, price\_id, product\_id

FROM coming

WHERE product\_id = ?

) AS p

ON r.id = p.price\_id

) AS pr

ON ca.product\_id = pr.product\_id

AND ca.outlet\_id = pr.outlet\_id;

-- актуальные сведения об объеме и ценах на указанный товар

-- по торговым точкам указанного типа

SELECT ca.product\_id, ca.coming\_amount - sa.selling\_amount AS amount, pr.item\_price

FROM (SELECT product\_id, SUM(amount) AS coming\_amount

FROM coming

WHERE outlet\_id IN (SELECT id

FROM outlet

WHERE outlet\_type\_id = ?

)

AND product\_id = ?

) AS ca

LEFT JOIN (SELECT product\_id, SUM(amount) AS selling\_amount

FROM selling

WHERE outlet\_id IN (SELECT id

FROM outlet

WHERE outlet\_type\_id = ?

)

AND product\_id = ?

) AS sa

ON ca.product\_id = sa.product\_id

INNER JOIN (

SELECT p.product\_id, r.item\_price

FROM (SELECT id, item\_price

FROM price

WHERE relevance = TRUE

) AS r

INNER JOIN (SELECT price\_id, product\_id

FROM coming

WHERE outlet\_id IN (SELECT id

FROM outlet

WHERE outlet\_type\_id = ?

)

AND product\_id = ?

) AS p

ON r.id = p.price\_id

) AS pr

ON ca.product\_id = pr.product\_id;

-- актуальные сведения об объеме и ценах на указанный товар

-- по конкретной торговой точке

SELECT ca.product\_id, ca.coming\_amount - sa.selling\_amount AS amount, pr.item\_price

FROM (SELECT product\_id, SUM(amount) AS coming\_amount

FROM coming

WHERE outlet\_id = ?

AND product\_id = ?

) AS ca

LEFT JOIN (SELECT product\_id, SUM(amount) AS selling\_amount

FROM selling

WHERE outlet\_id = ?

AND product\_id = ?

) AS sa

ON ca.product\_id = sa.product\_id

INNER JOIN (

SELECT p.product\_id, r.item\_price

FROM (SELECT id, item\_price

FROM price

WHERE relevance = TRUE

) AS r

INNER JOIN (SELECT price\_id, product\_id

FROM coming

WHERE outlet\_id = ?

AND product\_id = ?

) AS p

ON r.id = p.price\_id

) AS pr

ON ca.product\_id = pr.product\_id;

1. **Получить данные о выработке на одного продавца за указанный период по всем торговым точкам, по торговым точкам заданного типа**

-- данные о выработке на одного продавца за указанный период

-- по всем торговым точкам

SELECT outlet\_id, AVG(count) AS output

FROM (SELECT outlet\_id, COUNT(\*) AS count

FROM selling

WHERE sale\_datetime > ?

AND sale\_datetime < ?

GROUP BY employee\_id, outlet\_id

) AS a

GROUP BY outlet\_id;

-- данные о выработке на одного продавца за указанный период

-- по торговым точкам заданного типа

SELECT outlet\_id, AVG(count) AS output

FROM (SELECT outlet\_id, COUNT(\*) AS count

FROM selling

WHERE sale\_datetime > ?

AND sale\_datetime < ?

AND outlet\_id IN (SELECT id

FROM outlet

WHERE outlet\_type\_id = ?

)

GROUP BY employee\_id, outlet\_id

) AS a

GROUP BY outlet\_id;

-- данные о выработке на одного продавца за указанный период

-- по конкретной торговой точке

SELECT outlet\_id, AVG(count) AS output

FROM (SELECT outlet\_id, COUNT(\*) AS count

FROM selling

WHERE sale\_datetime > ?

AND sale\_datetime < ?

AND outlet\_id = ?

GROUP BY employee\_id, outlet\_id

) AS a

GROUP BY outlet\_id;

1. **Получить данные о выработке отдельно взятого продавца определенной торговой точки за указанный период**

-- данные о выработке отдельно взятого продавца

-- определенной торговой точки за указанный период

SELECT COUNT(\*) AS output

FROM selling

WHERE sale\_datetime

> ?

AND sale\_datetime

< ?

AND outlet\_id = ?

AND employee\_id = ?;

1. **Получить данные об объеме продаж указанного товара за некоторый период по всем торговым точкам, по торговым точкам заданного типа, по конкретной торговой точке**

-- данные об объеме продаж указанного товара за некоторый период

-- по всем торговым точкам

SELECT outlet\_id, SUM(amount)

FROM selling

WHERE product\_barcode = ?

AND sale\_datetime > ?

AND sale\_datetime < ?

GROUP BY outlet\_id;

-- данные об объеме продаж указанного товара за некоторый период

-- по торговым точкам заданного типа

SELECT outlet\_id, SUM(amount)

FROM selling

WHERE product\_barcode = ?

AND sale\_datetime > ?

AND sale\_datetime < ?

AND outlet\_id IN (SELECT id

FROM outlet

WHERE outlet\_type\_id = ?

)

GROUP BY outlet\_id;

-- данные об объеме продаж указанного товара за некоторый период

-- по конкретной торговой точке

SELECT SUM(amount)

FROM selling

WHERE product\_barcode = ?

AND sale\_datetime > ?

AND sale\_datetime < ?

AND outlet\_id = ?;

1. **Получить данные о заработной плате продавцов по всем торговым точкам, по торговым точкам заданного типа, по конкретной торговой точке**

-- данные о заработной плате продавцов

-- по всем торговым точкам

SELECT outlet\_id, fullname, salary

FROM employee

-- указать должность "продавец"

WHERE position\_id = ?;

-- данные о заработной плате продавцов

-- по торговым точкам заданного типа

SELECT outlet\_id, fullname, salary

FROM employee

-- указать должность "продавец"

WHERE position\_id = ?

AND outlet\_id IN (SELECT id

FROM outlet

WHERE outlet\_type\_id = ?

);

-- данные о заработной плате продавцов

-- по конкретной торговой точке

SELECT outlet\_id, fullname, salary

FROM employee

-- указать должность "продавец"

WHERE position\_id = ?

AND outlet\_id = ?;

1. **Получить сведения о поставках определенного товара указанным поставщиком за все время поставок либо за некоторый период**

-- сведения о поставках определенного товара указанным поставщиком за все время

SELECT s.\*, d.warehouse\_id, d.delivery\_date, d.amount

FROM supply AS s

RIGHT JOIN (SELECT \*

FROM distribution

WHERE supply\_id IN (SELECT id

FROM supply

WHERE product\_barcode = ?

AND supplier\_id = ?

)

) AS d

ON s.id = d.supply\_id;

-- сведения о поставках определенного товара указанным поставщиком за некоторый период

SELECT s.\*, d.warehouse\_id, d.delivery\_date, d.amount

FROM supply AS s

RIGHT JOIN (SELECT \*

FROM distribution

WHERE supply\_id IN (SELECT id

FROM supply

WHERE product\_barcode = ?

AND supplier\_id = ?

)

AND delivery\_date >= ?

AND delivery\_date <= ?

) AS d

ON s.id = d.supply\_id;

1. **Получить данные об арендованных помещениях, о количестве и перечне сотрудников, работающих в помещениях указанного типа, в конкретном помещении, по всем арендованным помещениям**

-- данные об арендованных помещениях

SELECT \*

FROM outlet

WHERE id IN (SELECT DISTINCT outlet\_id

FROM payment

WHERE payment\_type\_id = ? -- указать тип "аренда"

);

-- перечень сотрудников, работающих в помещениях указанного типа

SELECT DISTINCT fullname

FROM employee

WHERE outlet\_id IN (SELECT id

FROM outlet

WHERE outlet\_type\_id = ?

);

-- общее число сотрудников, работающих в помещениях указанного типа

SELECT COUNT(DISTINCT fullname)

FROM employee

WHERE outlet\_id IN (SELECT id

FROM outlet

WHERE outlet\_type\_id = ?

);

-- перечень сотрудников, работающих в конкретном помещении

SELECT DISTINCT fullname

FROM employee

WHERE outlet\_id = ?;

-- общее число сотрудников, работающих в конкретном помещении

SELECT COUNT(DISTINCT fullname)

FROM employee

WHERE outlet\_id = ?;

-- перечень сотрудников по всем арендованным помещениям

SELECT DISTINCT fullname

FROM employee

WHERE outlet\_id IN (SELECT id

FROM outlet

WHERE id IN (SELECT DISTINCT outlet\_id

FROM payment

WHERE payment\_type\_id = ? -- указать тип "аренда"

)

);

-- общее число сотрудников по всем арендованным помещениям

SELECT COUNT(DISTINCT fullname)

FROM employee

WHERE outlet\_id IN (SELECT id

FROM outlet

WHERE id IN (SELECT DISTINCT outlet\_id

FROM payment

WHERE payment\_type\_id = ? -- указать тип "аренда"

)

);

1. **Получить данные о рентабельности торговой точки: соотношение объема продаж к накладным расходам (суммарная заработная плата продавцов + платежи за аренду, коммунальные услуги) за указанный период**

-- данные о рентабельности торговой точки:

-- соотношение объема продаж к накладным расходам

-- (суммарная заработная плата продавцов + платежи за аренду,

-- коммунальные услуги) за указанный период

SELECT income / expenses

FROM (SELECT row\_number() OVER () as row\_number, SUM(price \* amount) AS income

FROM selling

WHERE outlet\_id = ?

AND sale\_datetime >= ?

AND sale\_datetime <= ?

) AS i

JOIN (SELECT row\_number() OVER () as row\_number, SUM(amount) AS expenses

FROM payment

WHERE outlet\_id = ?

AND datetime >= ?

AND datetime <= ?

) AS e

ON i.row\_number = e.row\_number;

1. **Получить сведения о поставках товаров по указанному номеру заказа**

-- сведения о поставках товаров по указанному номеру заказа

SELECT \*

FROM supply

WHERE id IN (SELECT DISTINCT supply\_id

FROM order\_item

WHERE order\_id IN (SELECT id

FROM orders

WHERE id = ?

)

);

1. **Получить сведения о покупателях указанного товара за обозначенный (или за весь) период по всем торговым точкам, по торговым точкам указанного типа, по данной торговой точке**

-- сведения о покупателях указанного товара за обозначенный

-- (или за весь) период по всем торговым точкам

SELECT \*

FROM client

WHERE id IN (SELECT DISTINCT client\_id

FROM selling

WHERE product\_barcode = ?

);

-- сведения о покупателях указанного товара за обозначенный

-- (или за весь) период по торговым точкам указанного типа

SELECT \*

FROM client

WHERE id IN (SELECT DISTINCT client\_id

FROM selling

WHERE product\_barcode = ?

AND outlet\_id IN (SELECT id

FROM outlet

WHERE outlet\_type\_id = ?

)

);

-- сведения о покупателях указанного товара за обозначенный

-- (или за весь) период по данной торговой точке

SELECT \*

FROM client

WHERE id IN (SELECT DISTINCT client\_id

FROM selling

WHERE product\_barcode = ?

AND outlet\_id = ?

);

1. **Получить сведения о наиболее активных покупателях по всем торговым точкам, по торговым точкам указанного типа, по данной торговой точке**

-- сведения о наиболее активных покупателях

-- по всем торговым точкам

SELECT \*

FROM client

WHERE id IN (SELECT client\_id

FROM selling

GROUP BY client\_id

ORDER BY SUM(amount) DESC

LIMIT 10

);

-- сведения о наиболее активных покупателях

-- по торговым точкам указанного типа

SELECT \*

FROM client

WHERE id IN (SELECT client\_id

FROM selling

WHERE outlet\_id IN (SELECT id

FROM outlet

WHERE outlet\_type\_id = ?

)

GROUP BY client\_id

ORDER BY SUM(amount) DESC

LIMIT 10

);

-- сведения о наиболее активных покупателях

-- по данной торговой точке

SELECT \*

FROM client

WHERE id IN (SELECT client\_id

FROM selling

WHERE outlet\_id = ?

GROUP BY client\_id

ORDER BY SUM(amount) DESC

LIMIT 10

);

1. **Получить данные о товарообороте торговой точки либо всех торговых точек определенной группы за указанный период**

-- данные о товарообороте торговой точки за указанный период

SELECT \*

FROM selling

WHERE sale\_datetime > ?

AND sale\_datetime < ?

AND outlet\_id = ?;

-- данные о товарообороте всех торговых точек определенной группы за указанный период

SELECT \*

FROM selling

WHERE sale\_datetime > ?

AND sale\_datetime < ?

AND outlet\_id IN (SELECT id

FROM outlet

WHERE outlet\_type\_id = ?

);