



МИНИСТЕРСТВО НАУКИ
И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное
образовательное учреждение высшего образования
«НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»



**НГТУ
НЭТИ** | **Факультет прикладной
математики и информатики**

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
Задание	3
Виды запросов в информационной системе	4
Основные понятия информационной системы	5
Структура организации	7
Бизнес-процессы торговой компании.....	9
2. Инфологическое проектирование.....	16
Список сущностей предметной области и их атрибуты	17
ER-диаграмма	23
3. Логическое проектирование	24
Скрипт для создания базы данных.....	30
4. Проектирование приложений	35
Цели, задачи и требования.....	35
Состав подсистем разрабатываемой ИС, назначение и функции каждого приложения ИС.....	36
Категории пользователей, работающих с приложениями ИС, выполняемые ими функции, права пользователей, категории данных, с которыми они работают	38
Отчеты	40
Запросы для формирования отчетов	40

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

- a. Старший продавец – создает заявки, принимает товары на торговой точке, подписывает акты о приемки товара.
- b. Продавец – реализует товар, выкладывает товар на прилавки.
- c. Уборщик – следит за чистотой на торговой точке.
- d. Директор – отвечает за согласование открытия/закрытия торговых точек и складов, подписывает все документы.
- e. Рекрутер – отвечает за подбор персонала.
- f. Бухгалтер – ведет финансовую отчетность компании.
- g. 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.

2. 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.

3. 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.

4. 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.

5. Склад

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

6. Заказ

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

7. Распределение

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

8. Заявка

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

9. Поступление

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

10. Цена

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

11. 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.

12. 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.

13. 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.

14. 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).

2. 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).

3. 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).

4. 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).

5. 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:

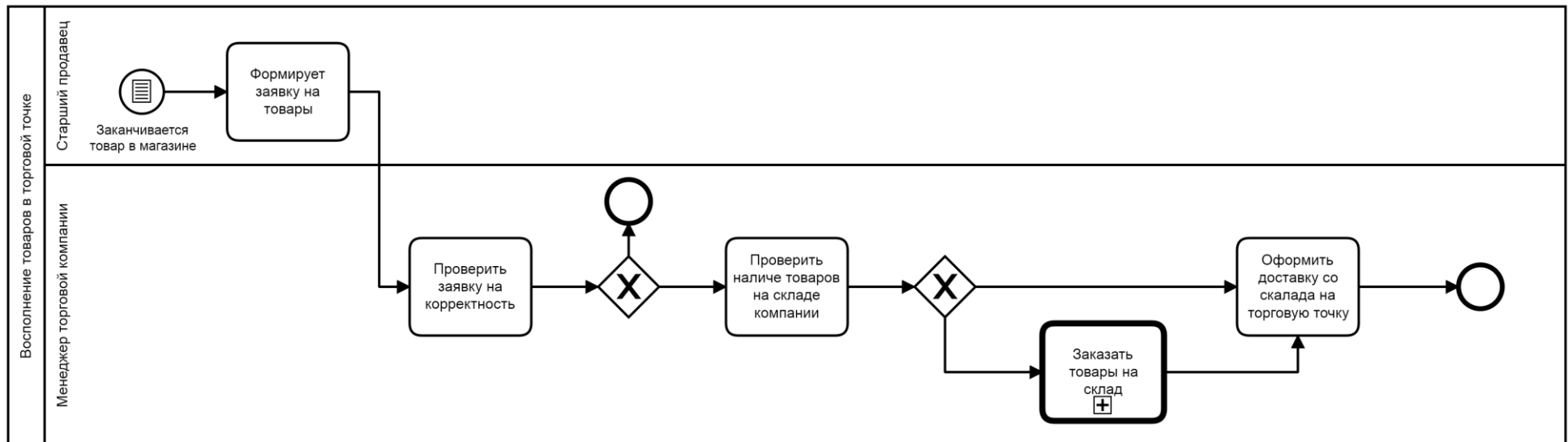
- Сведения о поставках (для экономического отдела).
- Акт о принятых и непринятых товарах на склад (я экономического отдела).

Бизнес-процессы торговой компании

1. Восполнение товаров в торговой точке

а. Старший продавец торговой точки составляет заявку на товары – фиксируются торговая точка, дата создания заявки, товары и заказанное количество товаров.

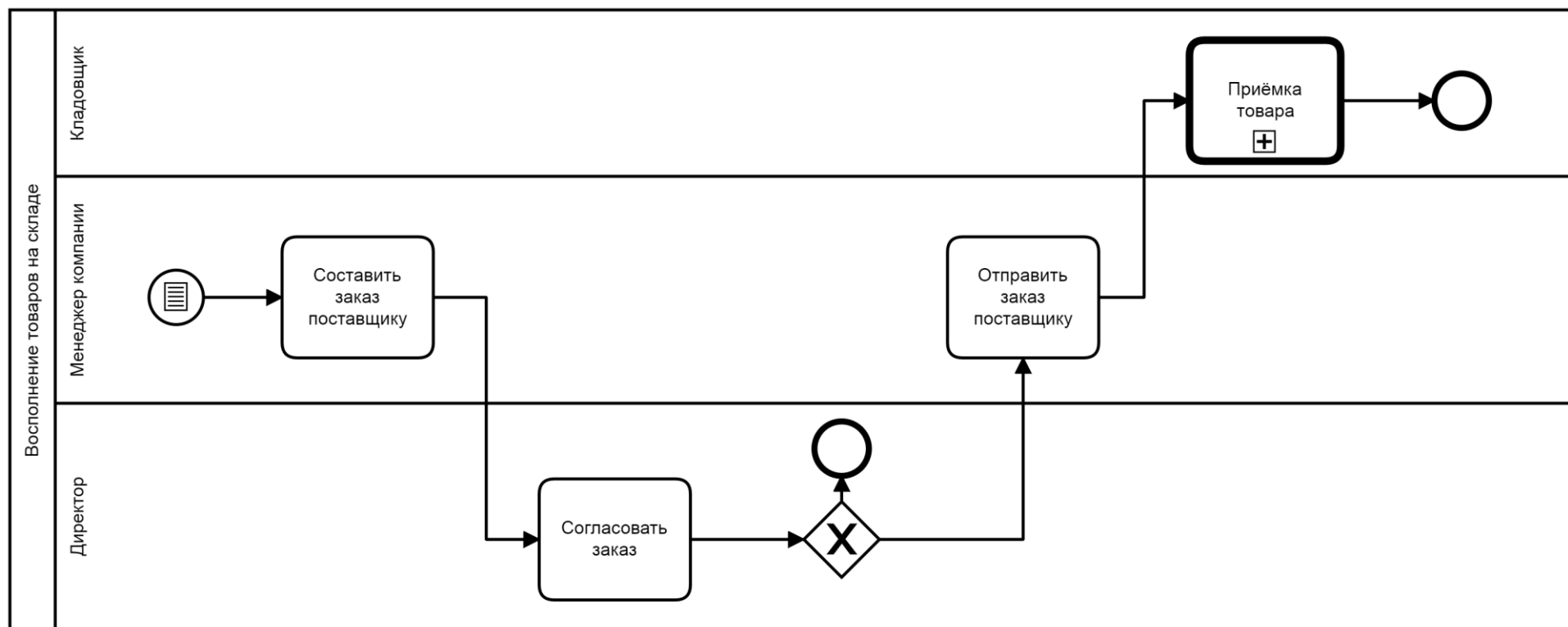
б. Менеджер торговой компании обрабатывает заявку – организует доставку товара в торговую точку и оформляет заказы на товары при их отсутствии на складе. При поступлении товаров в торговую точку фиксируются торговая точка, товары, дата поступления товаров, количество единиц товаров и розничная цена.



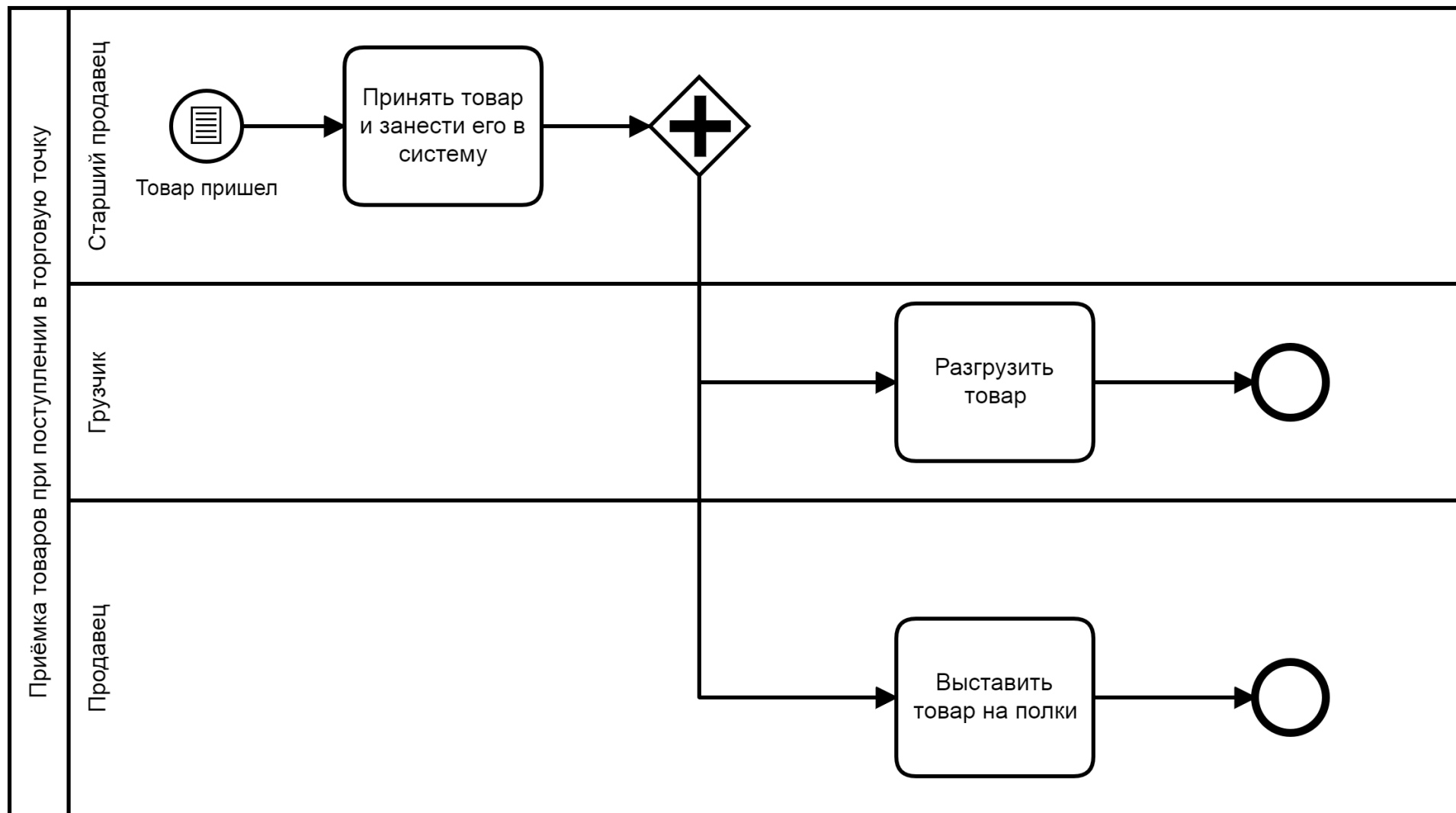
2. Восполнение товаров на складе

а. Менеджер торговой компании составляет заказ на поставки товаров – фиксируются склад, дата создания заказа, поставки и заказанное количество товаров.

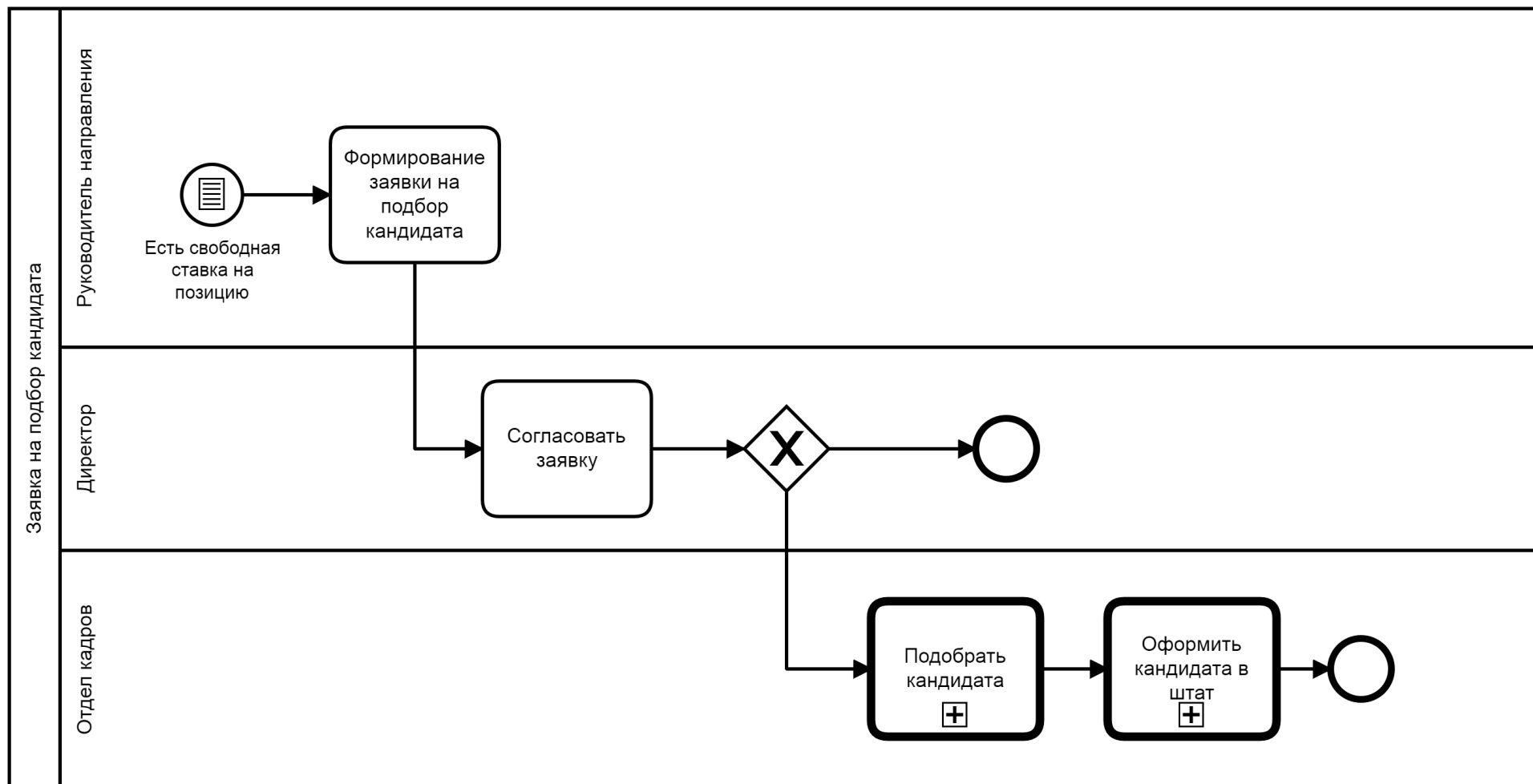
б. Поставщик исполняет обязательства по заказу – организует доставку товара на склад. При поступлении товаров на склад фиксируются склад, поставки, дата выполнения поставки и количество единиц товаров.



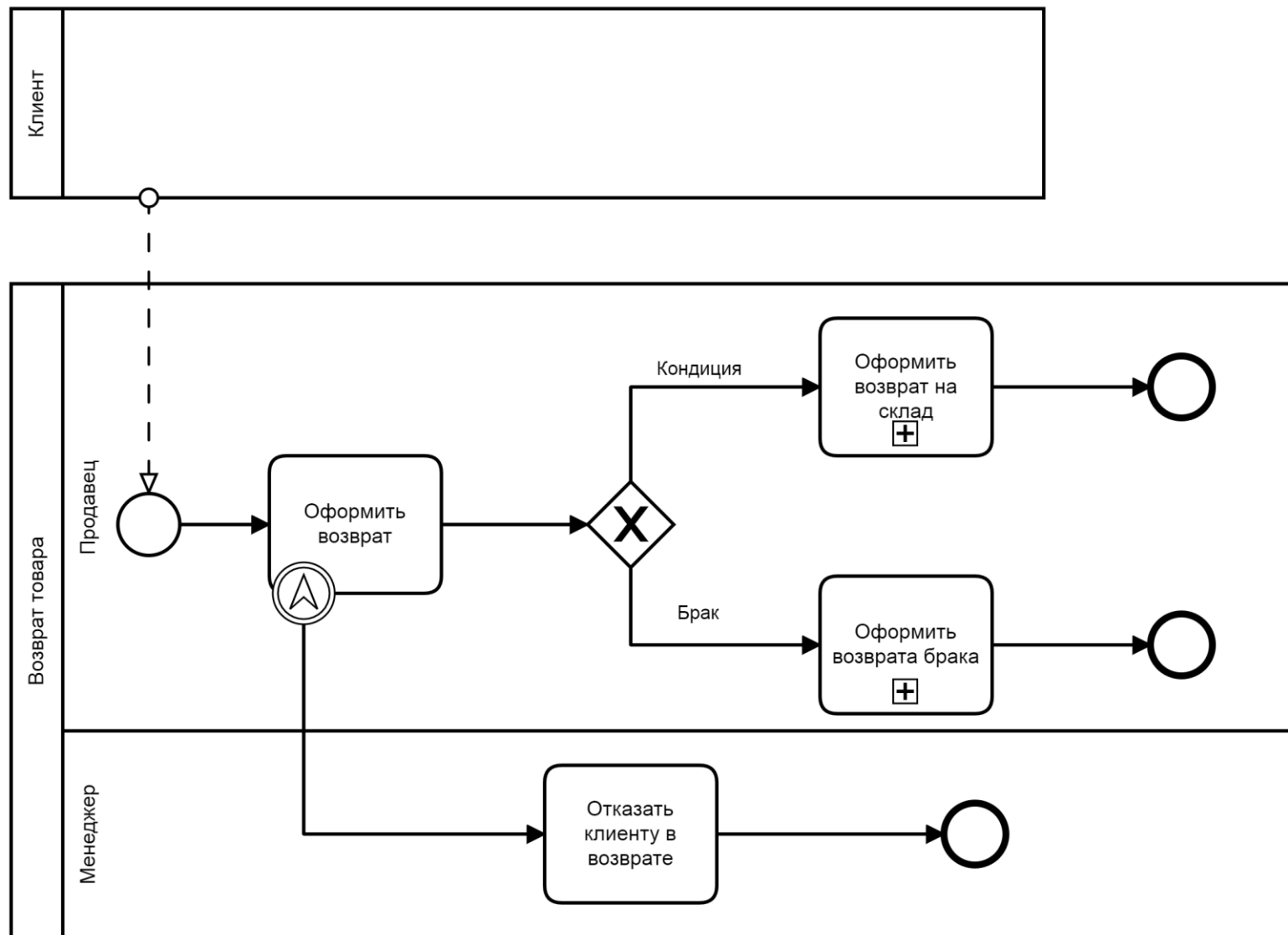
3. Приемка товаров при поступлении в торговую точку



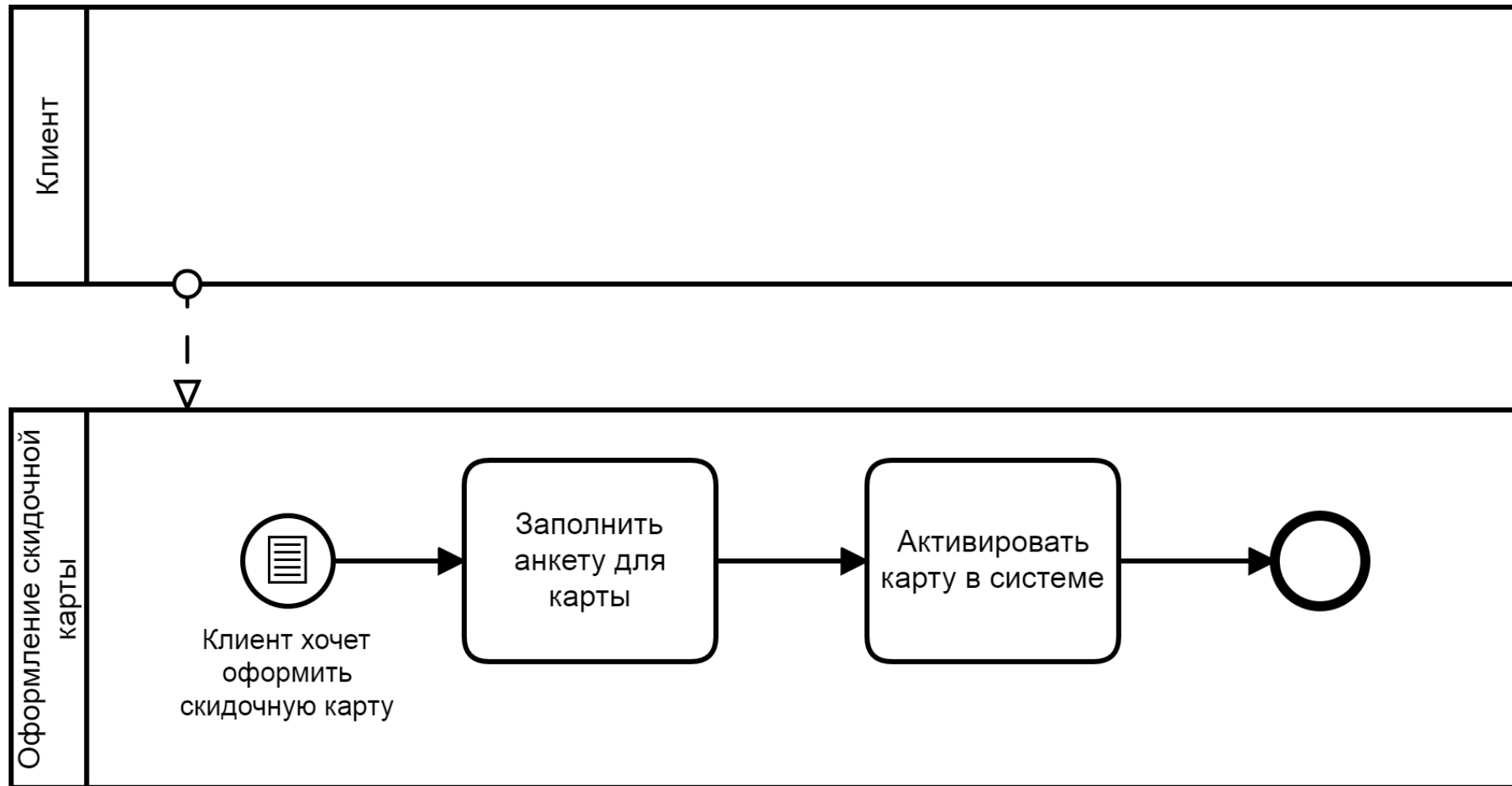
4. Заявка на подбор кандидата



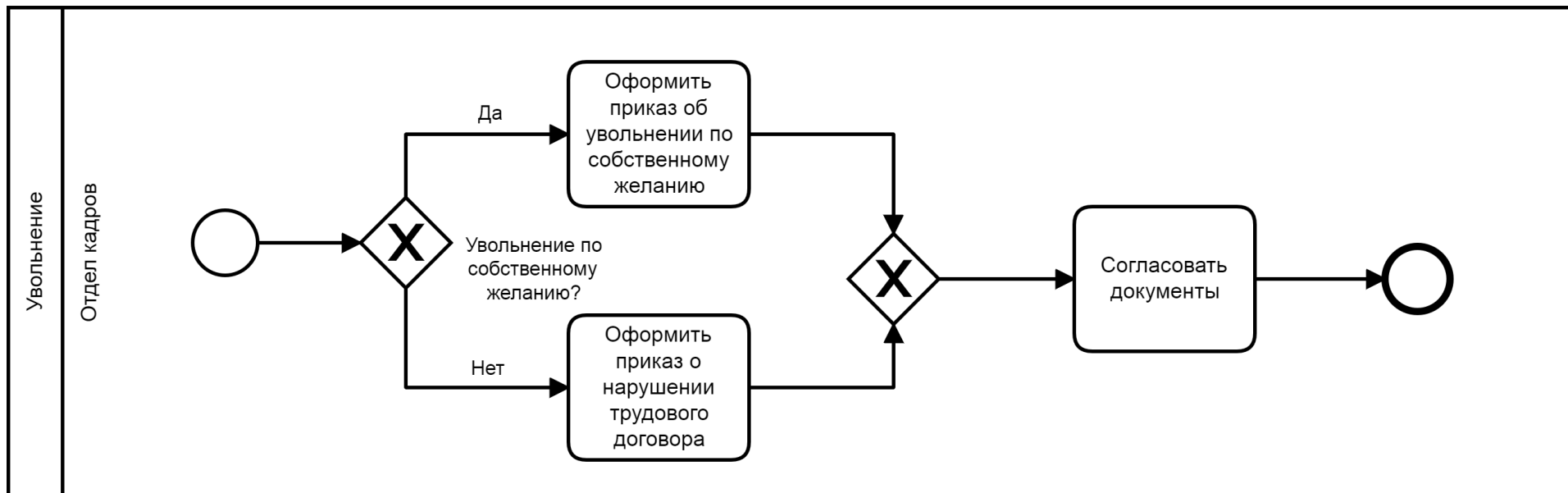
5. Возврат товара



6. Оформление скидочной карты



7. Увольнение



2. 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 обозначаящая 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 обозначаящая 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 обозначаящая 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

2. 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

3. Товар – стержневая сущность

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

4. Тип товара – обозначающая сущность

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

5. Позиция заказа – характеристическая сущность

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

6. Заказ – стержневая сущность

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

7. Заявка – стержневая сущность

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

8. Позиция заявки – характеристическая сущность

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

9. 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

10. 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

11. Поступление – ассоциативная сущность

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

12. Должность – обозначающая сущность

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

13. Торговая точка – стержневая сущность

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

14. Тип торговой точки – обозначающая сущность

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

15. 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

16. Цена – ассоциативная сущность

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

17. Распределение – ассоциативная сущность

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

18. Платеж – характеристическая сущность

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

19. Тип платежа – обозначающая сущность

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

20. Склад – стержневая сущность

- Адрес
- Площадь
- Дата открытия
- Дата закрытия

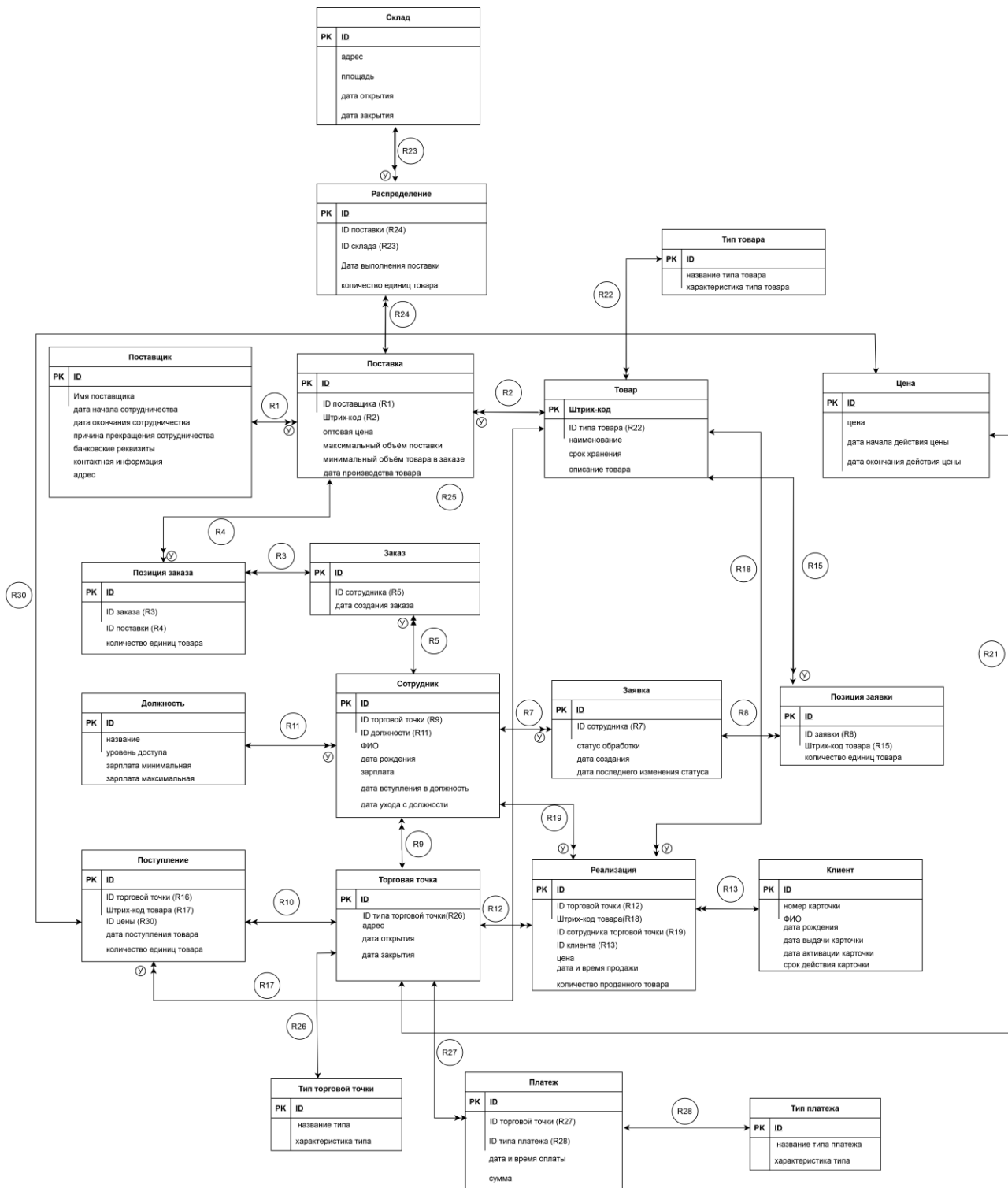
Множественность и условность связей между сущностями определены в таблице:

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:M	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.

Долж- ность	Сотруд- ник	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.
Сотруд- ник	Реализа- ция	1:Му	Реализация товара производится сотрудником (продавцом). Один продавец может реализовать множество товара. Также есть сотрудники, ни разу не реализовывавшие товар.
Торговая точка	Тип торго- вой точки	1:1	Каждая торговая точка относится к определённого типу.
Торговая точка	Сотруд- ник	1:М	Сотрудники работают в определенной торговой точке. В одной торговой точке может работать несколько сотрудников.
Торговая точка	Поступле- ние	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	Реализа- ция	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	Реализа- ция	1:М	Реализация составляется из покупок клиентов. Один клиент может купить множество товаров, в то время, как единица товара покупается конкретным клиентом.
Реализа- ция	Цена	1:1	Реализация товара происходит по конкретной актуальной цене.
Платеж	Тип пла- тежа	1:1	Каждый платеж относится к конкретному типу.

ER-диаграмма



3. 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	timestampz				
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	timestampz				
Заявка (application)	ID	id	integer				Генерация из последовательности
	ID сотрудника	employee_id	integer				
	статус обработки	status	character(20)				
	дата создания	creation_date	timestampz				

	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	timestampzt				
Цена (price)	ID	id	integer				Генерация из последовательности
	цена	item_price	money				Больше 0
	дата начала действия цены	begin_date	timestampzt				
	дата окончания действия цены	end_date	timestampzt				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	timestampzt				
	количество единиц товара	amount	integer				Больше 0
	ID	id	integer				Генерация из последовательности

Платеж (payment)	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      timestampz 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     timestampz 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        timestampz  not null,
    close_date       timestampz,
    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     timestampz,
    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      timestampz not null,
    amount        int,
    constraint amount_positive check (
        payment.amount > 0
    )
);

```

4. 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 использована 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 избежать 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 формирования 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 = ?;
```

2. 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 = ?
);
```

3. 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;
```

4. 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 sell-
ing_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
        ) 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
      ) AS pr
      ON ca.product_id = pr.product_id;

```

5. 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;
```

6. 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 = ?;
```

7. 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 = ?;
```

8. 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 = ?;
```

9. 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;
```

10. 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"
                                )
                    )
);

```


);

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

```
-- 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;
```

12. 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 = ?
                             )
           )
);
```

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

```
-- 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 = ?
              )
);
```

14. 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
);

```

15. 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 = ?
);

```