

# **Проектирование реляционной базы данных**

## **Тема: магазин настольных игр**

### **Содержание**

1. Инфологическое проектирование .....	2
1.1. Анализ предметной области .....	2
1.2. Анализ информационных задач и круга пользователей системы .....	3
2. Логическое проектирование .....	4
2.1. Преобразование ER-диаграммы в схему базы данных .....	4
2.2. Составление реляционных отношений.....	6
2.3. Нормализация полученных отношений (до 4НФ).....	9
2.4. Определение дополнительных ограничений целостности .....	13
2.5 Описание групп пользователей и прав доступа .....	13
3. Физическое проектирование .....	14
3.1 Создание таблиц .....	14
3.2. Создание представлений (готовых запросов) .....	17
3.3 Назначение прав доступа .....	19
3.4. Создание индексов.....	19
3.5. Разработка стратегии резервного копирования .....	20

# 1. Инфологическое проектирование

## 1.1. Анализ предметной области

База данных создается для обслуживания операций, проводящихся с товарами в магазине настольных игр. БД должна содержать данные о сотрудниках, производителях, поставках, самих товарах и продажах.

В соответствии с предметной областью система строится с учетом следующих особенностей:

- Каждый производитель может организовывать несколько поставок, но каждую поставку организует лишь один производитель.
- Поставка может иметь несколько разных товаров, а также несколько поставок могут содержать одинаковый товар.
- В одной продаже может реализоваться несколько товаров, также, как и один товар может реализовываться в нескольких продажах(если его хватает на складе).
- Один продавец одновременно может обслуживать только одну продажу, но всего может обслуживать сколько угодно.
- Заработная плата сотрудника зависит от занимаемой должности.
- Для каждого товара должно указываться количество экземпляров на складе.
- Директор магазина назначает старшего сотрудника.

Для создания ER-модели необходимо выделить сущности предметной области и указать их атрибуты:

1. **Производитель:** Атрибуты: ID производителя(ключевой), наименование производителя, адрес, телефон, сайт.
2. **Поставки:** Атрибуты: ID поставки(ключевой), количество товара, оптовая цена, дата.
3. **Товары:** Атрибуты: ID товара(ключевой), название товара, категория товара, количество на складе.
4. **Продажи:** ID продажи(ключевой), количество проданного товара, дата, время
5. **Продавцы:** Атрибуты: ID сотрудника(ключевой), ФИО, паспортные данные, телефон, электронная почта, должность.

Исходя из выявленных сущностей, построим ER–диаграмму:



## 1.2. Анализ информационных задач и круга пользователей системы

Определим группы пользователей, их запросы к базе данных:

### 1. Директор магазина:

- назначение старшего сотрудника;
- получение списка сотрудников;
- изменение информации о заработной плате;
- получение полной информации о товарах, поставках, продажах и производителях;
- изменение информации о поставках и производителях;
- архивирование данных о продажах, совершённых какое-то время назад.

### 2. Старший сотрудник:

- назначение задач сотрудникам;
- получение списка сотрудников;
- получение информации о товарах и продажах;
- внесение изменений в данные о продажах.

### 3. Отдела кадров:

- приём/увольнение сотрудников;
- внесение изменений в данные о сотрудниках

### 4. Бухгалтеры:

- выполнение операций по выплате зарплаты.

### 5. Продавцы:

- просмотр данных о сотрудниках;
- внесение изменений в данные о товарах;
- просмотр информации о товарах.

### 6. Посетители:

- Просмотр данных о товарах.

## 2. Логическое проектирование

### 2.1. Преобразование ER-диаграммы в схему базы данных

База данных создаётся на основании схемы базы данных. Преобразование ER-диаграммы в схему БД выполняется путем сопоставления каждой сущности и каждой связи, имеющей атрибуты, отношения (таблицы) БД. Связь типа 1:n (один-ко-многим) между отношениями реализуется через внешний ключ. Ключ вводится для дочернего отношения. Внешнему ключу должен соответствовать первичный или уникальный ключ основного (родительского) отношения.



Рис. 2. ER-диаграмма ПрО «Магазин настольных игр»

Связи реализация между ПРОДАЖАМИ и ТОВАРАМИ, а также содержание между ПОСТАВКАМИ и ТОВАРАМИ принадлежат к типу n:m (Рис. 2.). Этот тип связи реализуется через вспомогательные отношения Реализация и Содержание, которые содержат комбинации первичных ключей соответствующих исходных отношений.

Для схемы БД будем использовать следующие обозначения, представленные на рис. 3.

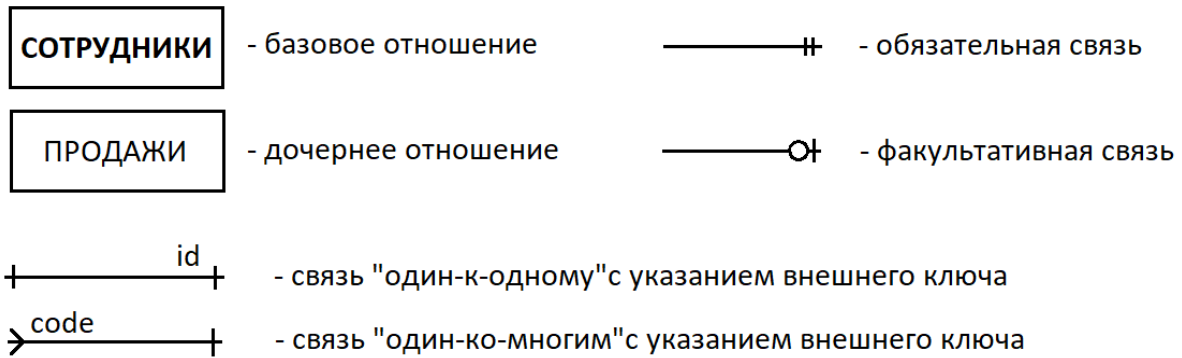


Рис. 3. Обозначения, используемые на схеме базы данных

Полученная схема реляционной базы данных (РБД) приведена на Рис. 4.

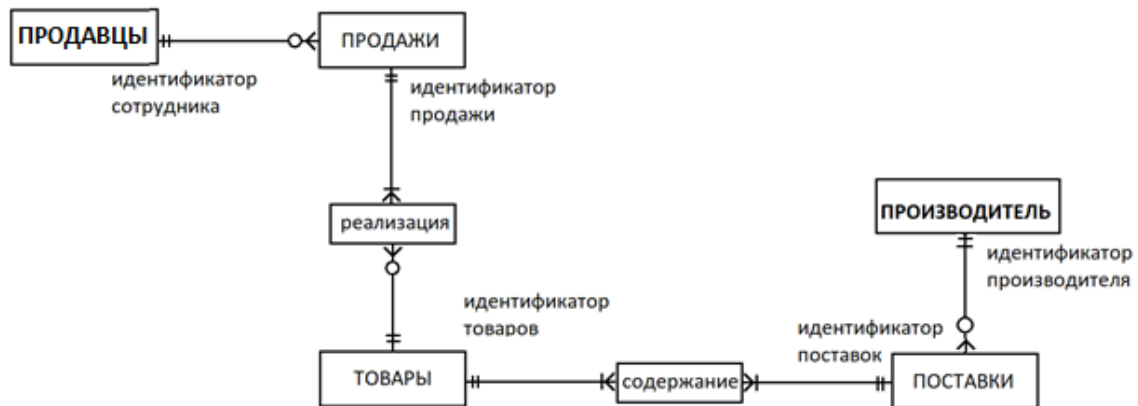


Рис. 4. Схема РБД, полученная из ER-диаграммы магазина настольных игр

## 2.2. Составление реляционных отношений

Каждое реляционное отношение соответствует одной сущности (объекту ПрО) и в него вносятся все атрибуты этой сущности. Для каждого отношения определяются первичный ключ и внешние ключи (в соответствии со схемой БД). В том случае, если базовое отношение не имеет потенциальных ключей, или если потенциальный ключ имеет большой размер, вводится **суррогатный первичный ключ**, который не несёт смысловой нагрузки и служит только для идентификации записей.

Отношения приведены в табл. 1-7. Для каждого отношения указаны атрибуты с их внутренним названием, типом и длиной. Типы данных обозначаются так: N – числовой, С – символьный тип фиксированной длины, V – символьный тип переменной длины, D – дата (этот тип имеет стандартную длину, зависящую от СУБД, поэтому она не указывается), Т – время.

Потенциальным ключом отношения ПРОДАВЦЫ является поле Паспортные данные, которое может меняться. Введём суррогатный первичный ключ Номер сотрудника.

Таблица 1. Схема отношения ПРОДАВЦЫ (Employees)

Содержание поля	Имя поля	Тип, длина	Примечания
Номер	ID	N (5)	<b>суррогатный первичный ключ</b>
ФИО	FIO	V (50)	обязательное поле
Паспортные данные	PASP	V (50)	обязательное уникальное поле
Должность	POS	V (30)	обязательное поле
Зар. плата	SAL	N (8,2)	обязательное поле, > 16259 руб.
Телефон	PHONE	V (30)	многозначное поле
Электронная почта	EMAIL	V (30)	многозначное поле

Потенциальными ключами отношения ТОВАРЫ являются поля Название и Артикул. Второе занимает меньше места, поэтому мы выберем его в качестве первичного ключа.

Таблица 2. Схема отношения ТОВАРЫ (Goods)

Содержание поля	Имя поля	Тип, длина	Примечания
Артикул	ID	N (5)	<b>первичный ключ</b>
Название	NAME	V (50)	обязательное поле
Категория	CAT	V (20)	обязательное поле
Розничная цена	PRC	N (8,2)	обязательное поле, > 0
Кол-во на складе	AMT	N (3)	обязательное поле

Потенциальным ключом вспомогательного отношения СОДЕРЖАНИЕ является комбинация первых двух полей этого отношения.

Таблица 3. Схема отношения СОДЕРЖАНИЕ (Content)

Содержание поля	Имя поля	Тип, длина	Примечания	
Поставка	SID	N (5)	внешний ключ (к Supply)	составной первичный ключ
Товар	GID	N (5)	внешний ключ (к Goods)	
Цена за единицу	PRC	N (8,2)	обязательное поле, > 0	
Количество товара	AMT	N(3)	обязательное поле, >0	

Для удобства ссылки на отношение ПРОДАЖИ, лучше завести суррогатный первичный ключ.

Таблица 4. Схема отношения ПРОДАЖИ (Sales)

Содержание поля	Имя поля	Тип, длина	Примечания
Номер	ID	N (5)	<b>суррогатный первичный ключ</b>
Отв-й сотрудник	EID	N (5)	внешний ключ (к Employees)
Время	TIMG	T	обязательное поле
Дата	DATG	D	обязательное поле

Для удобства введём суррогатный первичный ключ для отношения ПОСТАВКИ.

Таблица 5. Схема отношения ПОСТАВКИ (Supply)

Содержание поля	Имя поля	Тип, длина	Примечания
Номер	ID	N (5)	<b>суррогатный первичный ключ</b>
Производитель	PID	N (5)	внешний ключ (к Producer)
Кол-во товара	AMT	N (3)	обязательное поле, > 0
Оптовая цена	PRC	N (8,2)	обязательное поле, > 0
Дата	DATG	D	обязательное поле

Потенциальным ключом отношения ПРОИЗВОДИТЕЛИ является поле Наименование, но для удобства и экономии размера введём суррогатный первичный ключ.

Таблица 6. Схема отношения ПРОИЗВОДИТЕЛЬ (Producer)

Содержание поля	Имя поля	Тип, длина	Примечания
Номер производителя	ID	N (5)	<b>суррогатный первичный ключ</b>
Наименование	NAME	V (50)	обязательное поле
Адрес	ADR	V (50)	обязательное поле
Телефон	PHONE	V (30)	обязательное многозначное поле
Сайт	WEB	V (30)	

Потенциальным ключом вспомогательного отношения РЕАЛИЗАЦИЯ является комбинация первых двух полей этого отношения.

Таблица 7. Схема отношения РЕАЛИЗАЦИЯ (Realisation)

Содержание поля	Имя поля	Тип, длина	Примечания	
Продажа	SID	N (5)	внешний ключ (к Sales)	составной первичный ключ
Товар	GID	N (5)	внешний ключ (к Goods)	
Тип оплаты	WAY	V (8)	обязательное поле	
Цена	PRC	N (8,2)	обязательное поле, > 0	
Количество	AMT	N (3)	обязательное поле, > 0	



## 2.3. Нормализация полученных отношений (до 4НФ)

**1 НФ.** Для приведения таблиц к 1НФ требуется составить прямоугольные таблицы (одно значение атрибута – одна ячейка таблицы) и разбить сложные атрибуты на простые.

В отношении СОТРУДНИКИ разделим атрибут ФИО на два атрибута Фамилия и Имя, отчество, Паспортные данные на Номер паспорта (уникальный), Дата выдачи и Кем выдан.

Многозначные атрибуты Электронная почта и Телефон из отношения СОТРУДНИКИ вынесем в отдельное отношение ПОЧТА-ТЕЛЕФОН. В данном отношении нет потенциальных ключей: оставим его без первичного ключа, т. к. на него никто не ссылается.

В отношении ПРОДАЖИ разделим атрибут Время на два атрибута Время продажи и Время возврата, а также атрибут Дата на два атрибута Дата продажи и Дата возврата и создадим отдельную таблицу ВОЗВРАТЫ для атрибутов Время возврата и Дата возврата.

В отношении ПОСТАВКИ разделим атрибут Дата на два атрибута Дата отправки и Дата прибытия.

В отношении ПРОИЗВОДИТЕЛЬ разделим атрибут Адрес на два атрибута Юридический адрес и Адрес склада.

В отношении ТОВАРЫ имеет смысл создать для атрибута Категория отдельную справочную таблицу.

В отношении РЕАЛИЗАЦИЯ имеет смысл создать для атрибута Тип оплаты отдельную справочную таблицу.

**2 НФ.** В нашем случае составные первичные ключи имеют отношения СОДЕРЖАНИЕ и РЕАЛИЗАЦИЯ. Неключевые атрибуты этих отношений функционально полно зависят от составных первичных ключей.

**3 НФ.** В отношении СОТРУДНИКИ атрибут Зар. плата зависит от атрибута Должность, а не от первичного ключа. Поэтому создадим новое отношение ДОЛЖНОСТИ и перенесём в него атрибуты Должность и Зар. плата, а первичным ключом сделаем название должности.

В отношении ПОСТАВКИ атрибут Оптовая цена зависит от атрибута Количество товара. Поступим следующим образом: создадим новое отношение РАСЦЕНКА и вынесем туда атрибуты Количество товара и Оптовая цена. Стоимость товара будет исходить из приобретаемого количества. Целесообразно в качестве внешнего ключа сделать артикул товара.

**4 НФ.** Отношение ПОЧТА-ТЕЛЕФОН нарушают 4НФ, т. к. не всякая почта привязана к конкретному телефону(т. е. мы имеем две многозначных зависимости в одном отношении). Но выделять Телефоны в отдельное отношение не стоит, т. к. эти сведения носят справочный характер и не требуется их автоматическая обработка.

Отношения, полученные после нормализации приведены в табл. 7-20.

Таблица 8. Схема отношения ПРОДАВЦЫ (Employees)

Содержание поля	Имя поля	Тип, длина	Примечания
Номер	ID	N (5)	<b>суррогатный первичный ключ</b>
Фамилия	FAM	V (30)	обязательное поле
Имя, Отчество	NAME	V (30)	обязательное поле
Серия и номер паспорта	PASP	C (10)	обязательное уникальное поле
Дата выдачи	DATE	D	обязательное поле
Кем выдан	OFF	V (50)	обязательное поле
Должность	POS	V (30)	внешний ключ (к Posts)

Таблица 9. Схема отношения ТОВАРЫ (Goods)

Содержание поля	Имя поля	Тип, длина	Примечания
Артикул	ID	N (5)	<b>первичный ключ</b>
Название	NAME	V (50)	обязательное поле
Категория	CAT	V (30)	внешний ключ (к Category)
Розничная цена	PRC	N (8,2)	обязательное поле, > 0
Кол-во на складе	AMT	N (3)	обязательное поле

Таблица 10. Схема отношения СОДЕРЖАНИЕ (Content)

Содержание поля	Имя поля	Тип, длина	Примечания	
Поставка	SID	N (5)	внешний ключ (к Supply)	составной первичный ключ
Товар	GID	N (5)	внешний ключ (к Goods)	
Цена за единицу	PRC	N (8,2)	обязательное поле, > 0	
Количество товара	AMT	N(3)	обязательное поле, >0	

Таблица 11. Схема отношения ПРОДАЖИ (Sales)

Содержание поля	Имя поля	Тип, длина	Примечания
Номер	ID	N (5)	<b>суррогатный первичный ключ</b>
Отв-й сотрудник	EID	N (5)	внешний ключ (к Employees)
Время продажи	TIMS	T	обязательное поле
Дата продажи	DATS	D	обязательное поле

Таблица 12. Схема отношения ПОСТАВКИ (Supply)

Содержание поля	Имя поля	Тип, длина	Примечания
Номер	ID	N (5)	<b>суррогатный первичный ключ</b>
Производитель	PID	N (5)	внешний ключ (к Producer)
Дата отправки	DATD	D	обязательное поле
Дата прибытия	DAT_A	D	

Таблица 13. Схема отношения ПРОИЗВОДИТЕЛЬ (Producer)

Содержание поля	Имя поля	Тип, длина	Примечания
Номер производителя	ID	N (5)	<b>суррогатный первичный ключ</b>
Наименование	NAME	V (50)	обязательное поле
Адрес юридический	ADRY	V (50)	обязательное поле
Адрес склада	ADRS	V (50)	обязательное поле
Телефон	PHONE	V (30)	обязательное многозначное поле
Сайт	WEB	V (30)	

Таблица 14. Схема отношения РЕАЛИЗАЦИЯ (Realisation)

Содержание поля	Имя поля	Тип, длина	Примечания		
Продажа	SID	N (5)	внешний ключ (к Sales)	составной первичный ключ	
Товар	GID	N (5)	внешний ключ (к Goods)		
Тип оплаты	WAY	V (15)	внешний ключ (к PMTD)		
Цена	PRC	N (8,2)	обязательное поле, > 0		
Количество	AMT	N (3)	обязательное поле, > 0		

Таблица 15. Схема отношения ПОЧТА-ТЕЛЕФОН (EmPh)

Содержание поля	Имя поля	Тип, длина	Примечания
Номер сотрудника	ID	N (5)	внешний ключ (к Employees)
Электронный адрес	ADR	V (30)	
Телефон	PHN	V (30)	

Таблица 16. Схема отношения ДОЛЖНОСТИ (Posts)

Содержание поля	Имя поля	Тип, длина	Примечания
Название должности	POST	V (30)	<b>первичный ключ</b>
Зар. плата	SAL	N (8,2)	обязательное поле, > 16259 руб.

Таблица 17. Схема отношения РАСЦЕНКА (Pricing)

Содержание поля	Имя поля	Тип, длина	Примечания
Номер товара	ID	N (5)	внешний ключ (к Goods)
Кол-во товара	AMT	N (3)	обязательное поле, > 0
Оптовая цена	PRICE	N (8,2)	обязательное поле, > 0

Таблица 18. Схема отношения КАТЕГОРИЯ (Category)

Содержание поля	Имя поля	Тип, длина	Примечания
Название	NAME	V(30)	<b>Первичный ключ</b>

Таблица 19. Схема отношения ВОЗВРАТЫ (Refunds)

Содержание поля	Имя поля	Тип, длина	Примечания
Номер продажи	ID	N (5)	внешний ключ (к Sales)
Дата возврата	DATR	D	
Время возврата	TIMR	T	

Таблица 20. Схема отношения ТИП ОПЛАТЫ (PMTD)

Содержание поля	Имя поля	Тип, длина	Примечания
Название	WAY	V (15)	<b>Первичный ключ</b>

Схема базы данных после нормализации приведена на рис. 5.

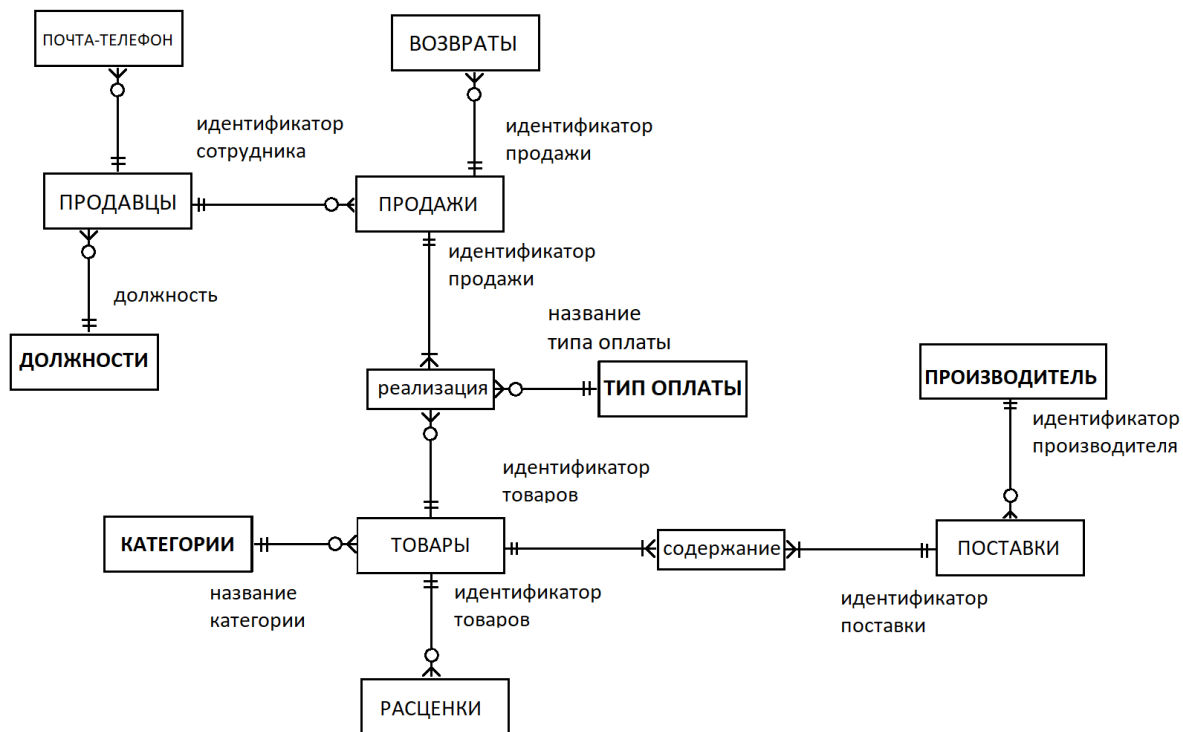


Рис. 5. Окончательная схема БД магазина настольных игр

## 2.4. Определение дополнительных ограничений целостности

В поле Количество на складе хранится общая сумма единиц товара на складе. Значение поля больше либо равно 0.

## 2.5 Описание групп пользователей и прав доступа

Опишем для каждой группы пользователей права доступа к каждой таблице. Права доступа должны быть распределены так, чтобы для каждого объекта БД был хотя бы один пользователь, который имеет право добавлять и удалять данные из объекта. Права приведены в табл. 21. Используются следующие сокращения:

- s – чтение данных (select);
- i – добавление данных (insert);
- u – модификация данных (update);
- d – удаление данных (delete).

Права назначает администратор БД (или администратор безопасности, если система сложная и администраторов несколько).

Таблица 21. Права доступа к таблицам для групп пользователей.

Таблицы	Группы пользователей (роли)					
	Директор магазина	Старший сотрудник	Отдел кадров	Бухгалтеры	Продавцы	Посетители
Сотрудники	S	SIUD	SIUD	S	S	
Товары	SIUD	SIUD		S	US	S
Содержание	S	SIUD		S		
Продажи	S	SIUD		S	SIUD	
Поставки	SIUD	S		S		
Производитель	SIUD	S				
Реализация	S	SIUD		S	SIUD	
Почта-Телефон	S	S	SIUD	S	S	
Должности	S	S	SIUD	S		
Расценка	SIUD			S		
Категория	S	SIUD		S	SIUD	S
Возвраты	S	SIUD		S	SIUD	
Тип оплаты	SIUD	S		S	S	S

## 3. Физическое проектирование

### 3.1 Создание таблиц

#### 1. Отношение Employees (сотрудники):

```
CREATE TABLE Employees (  
    id NUMERIC(5) PRIMARY KEY,  
    fam VARCHAR(30) NOT NULL,  
    name VARCHAR(30) NOT NULL,  
    pasp CHAR(10) NOT NULL UNIQUE,  
    dat DATE NOT NULL,  
    off VARCHAR(50) NOT NULL,  
    pos VARCHAR(30) NOT NULL,  
    FOREIGN KEY (pos) REFERENCES Posts(post)  
) ENGINE = InnoDB;
```

#### 2. Отношение Goods (товары):

```
CREATE TABLE Goods (  
    id NUMERIC(5) PRIMARY KEY,  
    name VARCHAR(50) NOT NULL,  
    cat VARCHAR(30) NOT NULL,  
    prc NUMERIC(8,2) NOT NULL,  
    amt NUMERIC(3) NOT NULL,  
    CHECK (prc > 0),  
    CHECK (amt >= 0),  
    FOREIGN KEY (cat) REFERENCES Category(name)  
) ENGINE = InnoDB;
```

#### 3. Отношение Content (содержание):

```
CREATE TABLE Content (  
    sid NUMERIC(5),  
    gid NUMERIC(5),  
    prc NUMERIC(8,2) NOT NULL,  
    amt NUMERIC(3) NOT NULL,  
    CHECK (prc > 0),  
    CHECK (amt > 0),  
    PRIMARY KEY (sid, gid),  
    FOREIGN KEY (sid) REFERENCES Supply(id),  
    FOREIGN KEY (gid) REFERENCES Goods (id)  
) ENGINE = InnoDB;
```

#### 4. Отношение Sales (продажи):

```
CREATE TABLE Sales (  
    id NUMERIC(5) PRIMARY KEY,  
    eid NUMERIC(5) NOT NULL,  
    tims TIME NOT NULL,  
    datd DATE NOT NULL,  
    FOREIGN KEY (eid) REFERENCES Employees(id)  
) ENGINE = InnoDB;
```

#### 5. Отношение Supply (поставки):

```
CREATE TABLE Supply (  
    id NUMERIC(5) PRIMARY KEY,  
    pid NUMERIC(5) NOT NULL,  
    datd DATE NOT NULL,  
    dat_a DATE,  
    FOREIGN KEY (pid) REFERENCES Producer(id)  
) ENGINE = InnoDB;
```

#### 6. Отношение Producer (производитель):

```
CREATE TABLE Producer (  
    id NUMERIC(5) PRIMARY KEY,  
    name VARCHAR(50) NOT NULL,  
    adry VARCHAR(50) NOT NULL,  
    adrs VARCHAR(50) NOT NULL,  
    phone VARCHAR(30) NOT NULL,  
    web VARCHAR(30)  
) ENGINE = InnoDB;
```

#### 7. Отношение Realisation (реализация):

```
CREATE TABLE Realisation (  
    sid NUMERIC(5),  
    gid NUMERIC(5),  
    way VARCHAR(15) NOT NULL,  
    prc NUMERIC(8,2) NOT NULL,  
    amt NUMERIC(3) NOT NULL,  
    PRIMARY KEY (sid, gid),  
    FOREIGN KEY (way) REFERENCES PMTD(way),  
    CHECK (prc > 0),  
    CHECK (amt > 0)  
) ENGINE = InnoDB;
```

## 8. Отношение EmPh (почта-телефон):

```
CREATE TABLE EmPh (  
    id NUMERIC(5) NOT NULL,  
    adr VARCHAR(30),  
    phn VARCHAR(30),  
    FOREIGN KEY (id) REFERENCES Employees(id)  
) ENGINE = InnoDB;
```

## 9. Отношение Posts (должности):

```
CREATE TABLE Posts (  
    post VARCHAR(30) PRIMARY KEY,  
    sal NUMERIC(8,2) NOT NULL,  
    CHECK (sal > 16259)  
) ENGINE = InnoDB;
```

## 10. Отношение Pricing (расценка):

```
CREATE TABLE Pricing (  
    id NUMERIC(5) NOT NULL,  
    amt NUMERIC(3) NOT NULL,  
    price NUMERIC(8,2) NOT NULL,  
    CHECK (amt > 0),  
    CHECK (price > 0),  
    FOREIGN KEY (id) REFERENCES Goods (id)  
) ENGINE = InnoDB;
```

## 11. Отношение Category (категория):

```
CREATE TABLE Category (  
    name VARCHAR(30) PRIMARY KEY  
) ENGINE = InnoDB;
```

## 12. Отношение Refunds (возвраты):

```
CREATE TABLE Refunds (  
    id NUMERIC(5) NOT NULL,  
    datr DATE,  
    timr TIME,  
    FOREIGN KEY (id) REFERENCES Sales(id)  
) ENGINE = InnoDB;
```

## 13. Отношение PMTD (Тип оплаты):

```
CREATE TABLE PMTD (  
    way VARCHAR(15) PRIMARY KEY,  
) ENGINE = InnoDB;
```



## 3.2. Создание представлений (готовых запросов)

Приведём примеры нескольких готовых запросов (представлений):

### 1. Электронные адреса и телефоны сотрудников:

```
CREATE OR REPLACE VIEW emp_conts AS
SELECT e.name||' '||e.fam, p.adr, p.phn
FROM EmPh p, Employees e
WHERE e.ID=p.ID
ORDER BY e.name;
```

### 2. Процент покупок, оплаченных картой за сутки:

```
CREATE OR REPLACE VIEW Card_payments AS
SELECT (count(way)*100)/(SELECT count(tims) FROM Sales WHERE tims
BETWEEN (ADDTIME(curtime(), '-23:59:0') AND curtime())
FROM Realisation, Sales
WHERE way = 'карта' and sid=id and tims BETWEEN (ADDTIME(curtime(),
'-23:59:0') AND curtime()))
```

### 3. Ожидаемые товары:

```
CREATE OR REPLACE VIEW w_supp AS
SELECT g.*
FROM Supply s, Goods g, Content c
WHERE g.ID=c.GID AND s.ID=c.SID AND s.dat_a IS NULL
ORDER BY g.name;
```

### 4. Товары в наличии, оптовая цена которых не больше 3000:

```
CREATE VIEW available_goods AS
SELECT g.id, g.name, g.amt
FROM Goods g, Pricing p
WHERE g.amt > 0 and p.id=g.id and price < 3000
ORDER BY g.amt;
```

### 5. Производители, от которых прибыли поставки за последнюю неделю:

```
CREATE OR REPLACE VIEW w_prod AS
SELECT p.*
FROM Producer p, Supply s
WHERE s.PID=p.ID AND DAT_A > (DATE_SUB(CURDATE(), INTERVAL 7 DAY))
ORDER BY p.name;
```

## 6. Возвращённые товары за месяц:

```
CREATE OR REPLACE VIEW Refunded AS
SELECT g.name , s.*
FROM Sales s, Realisation r, Goods g, Refunds ref
WHERE datr IS NOT NULL and datr BETWEEN (currdate() AND
DATE_ADD(currdate(), INTERVAL 1 MONTH)) and ref.id = s.id and s.id = r.sid
and r.gid = g.id
ORDER BY datr;
```

## 7. Потенциальная прибыль с вида товара:

```
CREATE VIEW Profit AS
SELECT g.name, g.amt*g.prc - g.amt*p.price
FROM Goods g, Pricing p
WHERE g.amt>0 and g.id = p.id
ORDER BY g.name;
```

Для того, чтобы можно было работать с этими представлениями, соответствующим пользователям нужно назначить права доступа к представлениям. Эти права перечислены в таблице 22.

Таблица 22. Права доступа к представлениям.

Представления	Группы пользователей		
	Руководители	Продавцы	Посетители
Почта и телефоны сотрудников	S	S	
Доля покупок, совершенных картой	S	S	
Ожидаемые товары	S	S	S
Товары, оптовая цена которых меньше 3000	S		
Производители, от которых прибыли поставки за последнюю неделю	S		
Возвращенные товары за месяц	S	S	
Потенциальная прибыль	S		

### 3.3 Назначение прав доступа

Права доступа пользователей предоставляются с помощью команды GRANT. Рассмотрим для примера права сотрудника компании r\_user, который является сотрудником отдела кадров. Права доступа к отношениям Employees, EmPh могут быть описаны следующим образом:

```
grant SELECT, INSERT, UPDATE, DELETE ON Employees to r_user;  
grant SELECT, INSERT, UPDATE, DELETE ON EmPh to r_user;
```

Права доступа старших сотрудников к представлению (manager) к представлению Refunds могут быть описаны следующим образом:

```
grant SELECT, INSERT, UPDATE, DELETE ON Refunds to manager;
```

Если сотрудник не является старшим, он сможет только добавить или получить данные без права на их редактирование.

Права доступа продавцов(staff) к представлению emp\_conts могут быть описаны следующим образом:

```
grant SELECT ON emp_conts to staff;
```

Права доступа посетителей(visitor) к представлению w\_supp могут быть описаны следующим образом:

```
grant SELECT ON w_supp to visitor;
```

### 3.4. Создание индексов

Анализ готовых запросов показывает, что для повышения эффективности работы с данными необходимо создать индексы для всех внешних ключей. Приведём примеры создания индексов:

```
CREATE INDEX ind_posts ON Employess (pos);  
CREATE INDEX ind_cats ON Goods (cat);  
CREATE INDEX ind_eid ON Sales (eid);  
CREATE INDEX ind_pid ON Supply (pid);
```

### 3.5. Разработка стратегии резервного копирования

Интенсивность обновления разработанной базы данных низкая, поэтому для обеспечения сохранности вполне достаточно проводить полное резервное копирование БД раз в день (перед окончанием рабочего дня). Для разработанной БД нет необходимости держать сервер включенным круглосуточно, поэтому можно создать соответствующее задание операционной системы, которое будет автоматически запускаться перед выключением сервера.