

Лекция  
ER-моделирование баз  
данных

# Основные свойства таблиц в реляционных БД

- Каждое значение, содержащееся на пересечении строки и колонки, должно быть *атомарным* (то есть не расчленяемым на несколько значений).
-

# Основные свойства таблиц в реляционных БД

- Каждое значение, содержащееся на пересечении строки и колонки, должно быть *атомарным* (то есть не расчленяемым на несколько значений).
- **Значения данных в одной и той же колонке должны принадлежать к одному и тому же типу, доступному для использования в данной СУБД.**

# Основные свойства таблиц в реляционных БД

- Каждое значение, содержащееся на пересечении строки и колонки, должно быть *атомарным* (то есть не расчленяемым на несколько значений).
- Значения данных в одной и той же колонке должны принадлежать к одному и тому же типу, доступному для использования в данной СУБД.
- **Каждая запись в таблице уникальна, то есть в таблице не существует двух записей с полностью совпадающим набором значений ее полей.**

# Основные свойства таблиц в реляционных БД

- Каждое значение, содержащееся на пересечении строки и колонки, должно быть *атомарным* (то есть не расчленяемым на несколько значений).
- Значения данных в одной и той же колонке должны принадлежать к одному и тому же типу, доступному для использования в данной СУБД.
- Каждая запись в таблице уникальна, то есть в таблице не существует двух записей с полностью совпадающим набором значений ее полей.
- **Каждое поле имеет уникальное имя.**

# Основные свойства таблиц в реляционных БД

- Каждое значение, содержащееся на пересечении строки и колонки, должно быть *атомарным* (то есть не расчленяемым на несколько значений).
- Значения данных в одной и той же колонке должны принадлежать к одному и тому же типу, доступному для использования в данной СУБД.
- Каждая запись в таблице уникальна, то есть в таблице не существует двух записей с полностью совпадающим набором значений ее полей.
- Каждое поле имеет уникальное имя.
- **Последовательность полей в таблице**

# Основные свойства таблиц в реляционных БД

- Каждое значение, содержащееся на пересечении строки и колонки, должно быть *атомарным* (то есть не расчленяемым на несколько значений).
- Значения данных в одной и той же колонке должны принадлежать к одному и тому же типу, доступному для использования в данной СУБД.
- Каждая запись в таблице уникальна, то есть в таблице не существует двух записей с полностью совпадающим набором значений ее полей.
- Каждое поле имеет уникальное имя.
- Последовательность полей в таблице несущественна.

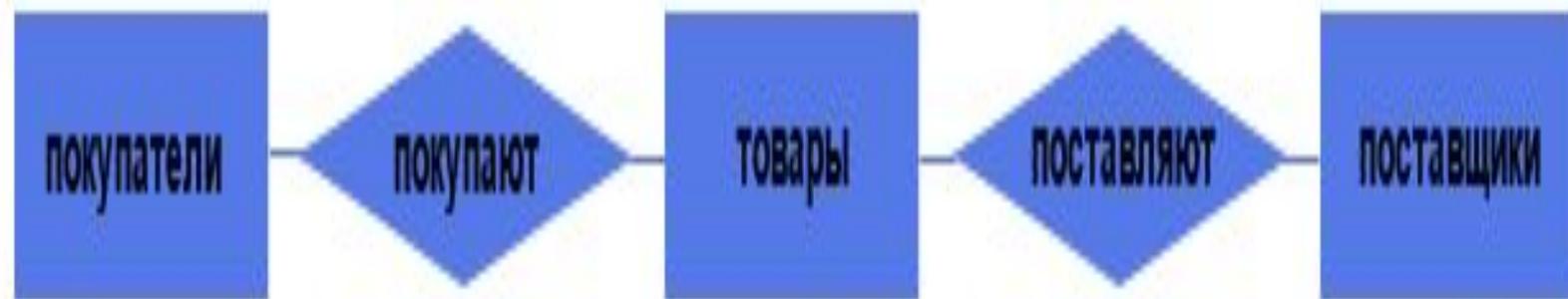
# Пример. Постановка задачи

- Предметная область - Интернет-магазин
- У магазина есть товары, которые поставляются поставщиками и покупаются покупатели.

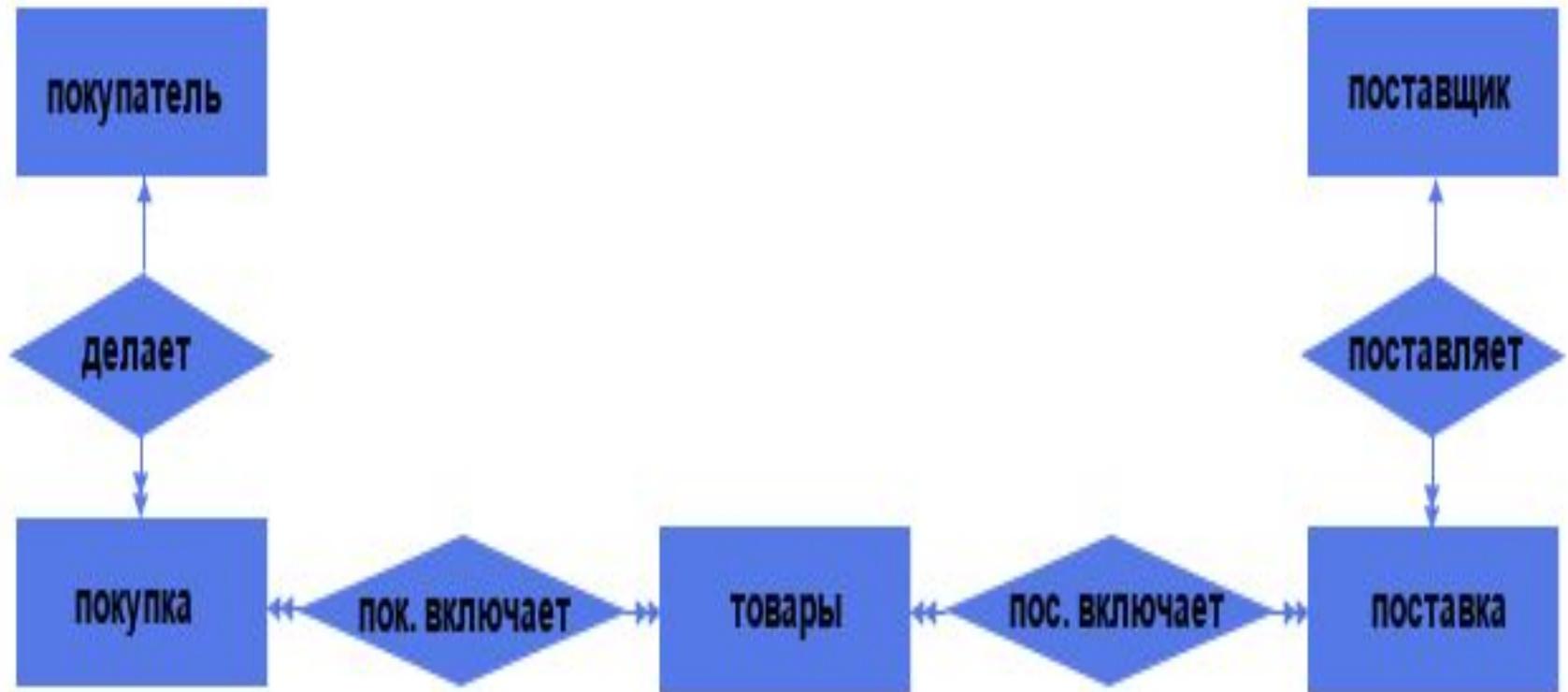
# **Концептуальная модель - это отражение предметной области, для которой разрабатывается база данных.**

Может быть представлена диаграммой с принятыми обозначениями элементов:

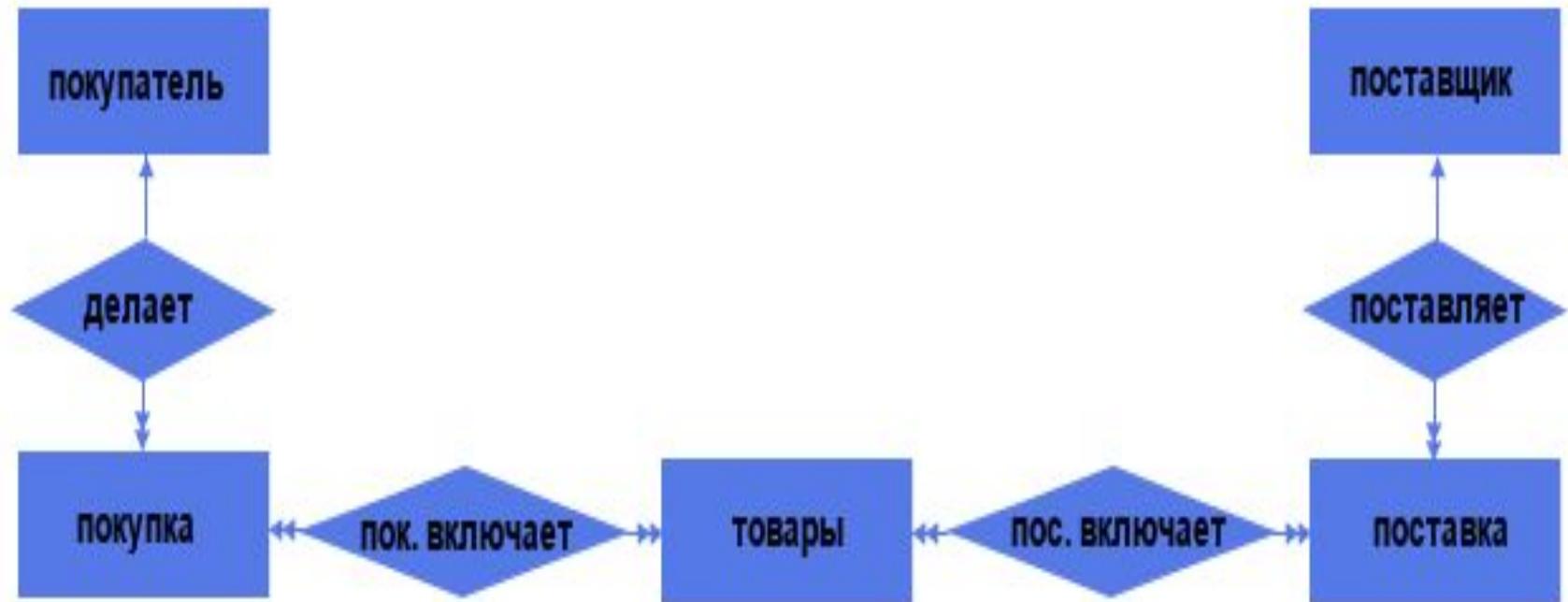
- все объекты, обозначающие объекты, представлены в виде прямоугольника;
- атрибуты, характеризующие объект - в виде овала;
- связи между объектами – ромбами;
- мощность связи обозначаются стрелками (в направлении, где мощность равна многим - двойная стрелка, а со стороны, где она равна единице - одинарная).



Учтем как поставщик поставляет  
товары, а покупатель покупает



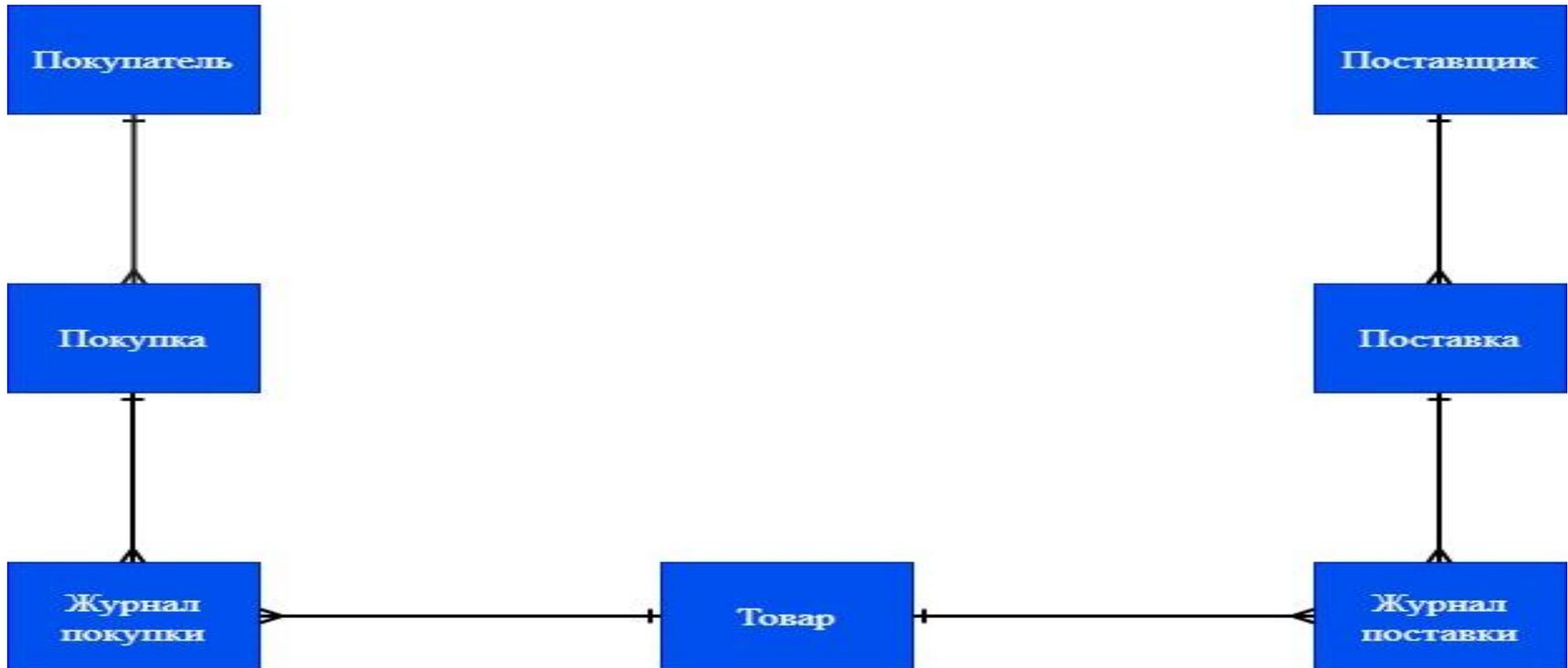
# Учтем как поставщик поставляет товары, а покупатель покупает



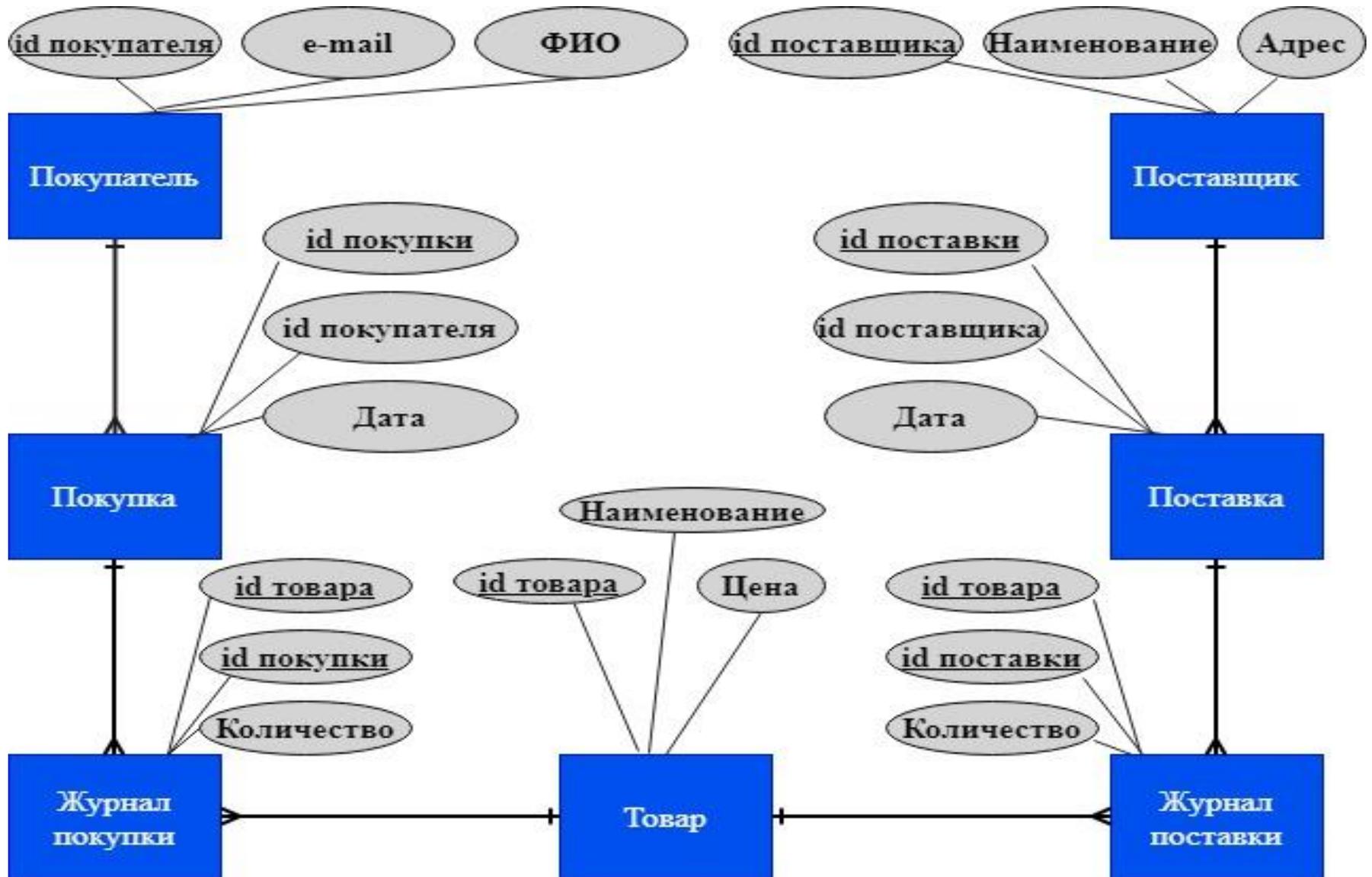
Две связи "многие ко многим" (для связи Поставка – Товар каждая поставка может содержать несколько товаров, а один и тот же товар может содержаться в нескольких

Используем в ER-диаграмме нотацию «вороньи лапки». Связи «один ко многим».

Обратите внимание, **в сильных сущностях** Покупатель, Товар, Поставщик **один экземпляр** сущности (на схеме один штришок) соответствует многим экземплярам связанных с ними **слабых сущностей** (на схеме «воронья лапка»).



# Концептуальная модель интернет-магазина



# Преобразование концептуальной модели в реляционную модель

- Построить набор предварительных таблиц и указать первичные ключи.
- Провести процесс нормализации.

В таблицах Журнал покупок и Журнал поставок составной первичный ключ из атрибутов, каждый из которых является внешним ключом к таблицам Товар, Покупка или Поставка. В таблицах Покупка и Поставка внешние ключи к таблицам Покупатель и Поставка.

Покупатель	Поставщик	Покупка	Поставка
Id покупателя (PK)	Id поставщика (PK)	Id покупки (PK)	Id поставки (PK)
ФИО	Наименование	Id покупателя (FK)	Id поставщика (FK)
E-mail	Адрес	Дата	Дата

Товар	Журнал покупок	Журнал поставок
Id товара (PK)	Id покупки (FK)	Id поставки (FK)
Наименование	Id товара (FK)	Id товара (FK)
Цена	Количество	Количество

} PK      } PK

PK – первичный ключ, FK – внешний ключ

*Первичный ключ* (сокращенно РК - primary key) - столбец, значения которого во всех строках различны.

Колонка, указывающая на запись в другой таблице, связанную с данной записью, называется **внешним ключом** (*foreign key*).

# Ищем неатомарные поля

Покупатель	Поставщик	Покупка	Поставка
Id покупателя (PK)	Id поставщика (PK)	Id покупки (PK)	Id поставки (PK)
ФИО	Наименование	Id покупателя (FK)	Id поставщика (FK)
E-mail	Адрес	Дата	Дата

Товар	Журнал покупок	Журнал поставок
Id товара (PK)	Id покупки (FK)	Id поставки (FK)
Наименование	Id товара (FK)	Id товара (FK)
Цена	Количество	Количество

PK

PK

Если будем учитывать поставки в определенные города, то из атрибута Адрес надо выделить Город в отдельный атрибут

<b>Поставщик</b>
Id поставщика (PK)
Наименование
Город
Адрес

Поле ФИО.

Если мы собираемся поздравлять покупателей с именинами (которые, как известно, зависят от имени), то это поле пришлось бы разбить на три:

- Фамилию,
- Имя
- Отчество.

Если магазин этого делать не собирается, то поле ФИО ***можно считать атомарным***, а таблицу Покупатели можно считать приведенной к 1НФ.

Так как в таблицах Покупатель, Поставщик, Покупка, Поставка, Товар первичный ключ простой.

А в таблицах Журнал поставок и Журнал покупок единственные неключевые атрибуты Количество функционально полно зависят от составных первичных ключей. (Количество зависит и от товара, и от того, в какой он покупке или поставке)

Покупатель	Поставщик	Покупка	Поставка
Id покупателя (PK)	Id поставщика (PK)	Id покупки (PK)	Id поставки (PK)
ФИО	Наименование	Id покупателя (FK)	Id поставщика (FK)
E-mail	Адрес	Дата	Дата

Товар	Журнал покупок	Журнал поставок
Id товара (PK)	Id покупки (FK)	Id поставки (FK)
Наименование	Id товара (FK)	Id товара (FK)
Цена	Количество	Количество

Diagram showing relationships between tables. A bracket labeled 'PK' groups the 'Id покупки (FK)' and 'Id товара (FK)' fields in the 'Журнал покупок' table. Another bracket labeled 'PK' groups the 'Id поставки (FK)' and 'Id товара (FK)' fields in the 'Журнал поставок' table.

## Таблица Товар.

- Добавим поле Дата (когда изменилась цена).
- Цена зависит от неключевого поля Дата, нужно избавиться от функциональной зависимости неключевых атрибутов.
- Преобразуем в 3НФ, введя дополнительную таблицу Цена.

Товар
Id товара (PK)
Наименование
Дата
Цена

Товар	Цена
Id товара (PK)	Id товара (FK)
Наименование	Дата
	Цена

} PK

# Нормализованная база Интернет-магазин в 3НФ

Покупатель
Id покупателя (PK)
ФИО
E-mail

Поставщик
Id поставщика (PK)
Наименование
Город
Адрес

Покупка
Id покупки (PK)
Id покупателя (FK)
Дата

Поставка
Id поставки (PK)
Id поставщика (FK)
Дата

Журнал покупок
Id покупки (FK)
Id товара (FK)
Количество

} PK

Журнал поставок
Id поставки (FK)
Id товара (FK)
Количество

} PK

Товар
Id товара (PK)
Наименование

Цена
Id товара (FK)
Дата
Цена

} PK