Проектирование реляционных БД на основе принципов нормализации

Этапы жизненного цикла базы данных

Процесс проектирования БД представляет собой последовательность переходов от неформального словесного описания информационной структуры предметной области к формализованному описанию объектов предметной области в терминах некоторой модели. В общем случае можно выделить следующие этапы проектирования:

Системный анализ и словесное описание информационных объектов предметной области.

Проектирование инфологической модели предметной области — частично формализованное описание объектов предметной области в терминах некоторой семантической модели, например, в терминах Е-модели.

Даталогическое или логическое проектирование БД, то есть описание БД в терминах принятой диалогической модели данных.

Физическое проектирование БД, то есть выбор эффективного размещения БД на внешних носителях для обеспечения наиболее эффективной работы приложения.

Если мы учтем, что между вторым и третьим этапами необходимо принять решение, с использованием какой стандартной СУБД будет реализовываться наш проект, то условно процесс проектирования БД можно представить последовательностью выполнения пяти соответствующих этапов

Системный анализ предметной области

С точки зрения проектирования БД в рамках системного анализа, необходимо осуществить первый этап, то есть провести подробное словесное описание объектов предметной области и реальных связей, которые присутствуют между описываемыми объектами. Желательно, чтобы данное описание позволяло корректно определить все взаимосвязи между объектами предметной области

В общем случае существуют два подхода к выбору состава и структуры предметной области:
Функциональный подход — он реализует принцип движения «от задач» и применяется тогда, когда заранее известны функции некоторой группы лиц и комплексов задач, для обслуживания информационных потребностей которых создается рассматриваемая БД. В этом случае мы можем четко выделить минимальный необходимый набор объектов предметной области, которые должны быть описаны.
Предметный подход — когда информационные потребности будущих пользователей БД жестко не фиксируются. Они могут быть многоаспектными и весьма динамичными. Мы не можем точно выделить минимальный набор объектов предметной области, которые необходимо описывать. В описание предметной области в этом случае включаются такие объекты и взаимосвязи, которые наиболее характерны и наиболее существенны для нее. БД, конструируемая при этом, называется предметной, то есть она может быть использована при решении множества разнообразных, заранее не определенных задач. Конструирование предметной БД в некотором смысле кажется гораздо более заманчивым, однако трудность всеобщего охвата предметной области с невозможностью конкретизации потребностей пользователей может привести к избыточно сложной схеме БД, которая для конкретных задач будет неэффективной.
На практике рекомендуется использовать некоторый компромиссный вариант, который, с одной стороны, ориентирован на конкретные задачи или функциональные потребности пользователей, а с другой стороны, учитывает возможность наращивания новых приложений.

Перед началом разработки необходимо иметь точное представление о том, что же должно выполняться в нашей системе, какие пользователи в ней будут работать, какие задачи будет решать каждый пользователь. И это правильно, ведь когда мы строим здание, мы тоже заранее предполагаем; для каких целей оно предназначено, в каком климате оно будет стоять, на какой почве, и в зависимости от этого проектировщики могут предложить нам тот или иной проект. Но, к сожалению, очень часто по отношению к базам данных считается, что все можно определить потом, когда проект системы уже создан. Отсутствие четких целей создания БД может свести на нет все усилия разработчиков, и проект БД получится «плохим», неудобным, не соответствующим ни реально моделируемому объекту, ни задачам, которые должны решаться с использованием данной БД.

Даталогическое проектирование

В реляционных БД даталогическое или логическое проектирование приводит к разработке схемы БД, то есть совокупности схем отношений, которые адекватно моделируют абстрактные объекты предметной области и семантические связи между этими объектами.
Основой анализа корректности схемы являются так называемые функциональные зависимости между атрибутами БД. Некоторые зависимости между атрибутами отношений являются нежелательными из-за побочных эффектов и аномалий, которые они вызывают при модификации БД.
В общем случае в результате выполнения этого этапа должны быть получены следующие результирующие документы:
Описание концептуальной схемы БД в терминах выбранной СУБД.
Описание внешних моделей в терминах выбранной СУБД.
Описание декларативных правил поддержки целостности базы данных.
Разработка процедур поддержки семантической целостности базы данных.
Однако перед тем как описывать построенную схему в терминах выбранной СУБД, нам надо корректно выстроить эту схему.

Корректной назовем схему БД, в которой отсутствуют нежелательные зависимости между атрибутами отношении.
Проектирование схемы БД может быть выполнено двумя путями:
путем декомпозиции (разбиения), когда исходное множество отношений, входящих в схему БД заменяется другим множеством отношений (число их при этом возрастает), являющихся проекциями исходных отношений;
путем синтеза, то есть путем компоновки из заданных исходных элементарных зависимостей между объектами предметной области схемы БД.
Классическая технология проектирования реляционных баз данных связана с теорией нормализации, основанной на анализе функций. Функциональные зависимости определяют устойчивые отношения между объектами и их свойствами в рассматриваемой предметной области функциональных зависимостей между атрибутами отношений

Нормализация

Процесс проектирования с использованием декомпозиции представляет собой процесс последовательной нормализации схем отношений, при этом каждая последующая итерация соответствует нормальной форме более высокого уровня и Каждой нормальной форме соответствует некоторый определенный набор ограничений, и отношение находится в некоторой нормальной форме, если удовлетворяет свойственному ей набору ограничений.

В теории реляционных БД обычно выделяется следующая последовательность нормальных форм:

первая нормальная форма (1NF);

вторая нормальная форма (2NF);

третья нормальная форма (3NF);

нормальная форма Бойса— Кодда (BCNF);

четвертая нормальная форма (4NF);

пятая нормальная форма, или форма проекции-соединения (5NF или PJNF).

Основные свойства нормальных форм:

- каждая следующая нормальная форма в некотором смысле улучшает свойства предыдущей;
- при переходе к следующей нормальной форме свойства предыдущих нормальных форм сохраняются.

Однако в процессе декомпозиции мы сталкиваемся с проблемой *обратимости,* то есть возможности восстановления исходной схемы. Таким образом, декомпозиция должна сохранять *эквивалентность* схем БД при замене одной схемы па другую.

*Схемы БД называются эквивалентными, если содержание исходной БД может быть получено путем естественного соединения отношений, входящих в результирующую схему, и п*ри выполнении эквивалентных преобразований сохраняется множество исходных фундаментальных функциональных зависимостей между атрибутами отношений.

При выполнении эквивалентных преобразований сохраняется множество исходных фундаментальных функциональных зависимостей между атрибутами отношений.

Функциональные зависимости определяют не текущее состояние БД, а все возможные ее состояния, то есть они отражают те связи между атрибутами, которые присущи реальному объекту, который моделируется с помощью БД

Основные определения

Функциональной зависимостью набора атрибутов B отношения R от набора атрибутов A того же отношения, обозначаемой как R.A -> R.B или A -> B называется такое соотношение проекций R[A] и R[B], при котором в каждый момент времени любому элементу проекции R[A] соответствует только один элемент проекции R[B] , входящий вместе с ним в какой-либо кортеж отношения R.

Функциональная зависимость R.A -> R.B называется *полной,* если набор атрибутов B функционально зависит от A и не зависит функционально от любого подмножества A,

то есть R.A -> R.B называется полной, если:

любое A1 с A=> R.A -/-> R.B, что читается следующим образом:

для любого A1, являющегося подмножеством A, R.B функционально не зависит от R.A, в противном случае зависимость R.A -> R.B называется неполной

Функциональная зависимость R.A -> R.B называется *транзитивной,* если существует набор атрибутов C такой, что:

1. C не является подмножеством A.
2. C не включает в себя B.
3. Существует функциональная зависимость R.A -> R.C.
4. Не существует функциональной зависимости R.C -> R.A.
5. Существует функциональная зависимость R.C -> R.B.

Возможным ключом отношения называется набор атрибутов отношения, который полностью и однозначно (функционально полно) определяет значения всех остальных атрибутов отношения, то есть возможный ключ — это набор атрибутов, однозначно определяющий кортеж отношения, и при этом при удалении любого атрибута из этого набора его свойство однозначной идентификации кортежа теряется.

Отношение — это подмножество декартова произведения множества доменов. Для каждого отношения всегда существует набор атрибутов, по которому можно однозначно определить кортеж отношения

Среди всех возможных ключей отношения обычно выбирают один, который считается главным и который называют *первичным ключом отношения.*

Неключевым атрибутом называется любой атрибут отношения, не входящий в состав ни одного возможного ключа отношения.

Взаимно-независимые атрибуты — это такие атрибуты, которые не зависят функционально один от другого.

Если в отношении существует несколько функциональных зависимостей, то каждый атрибут или набор атрибутов, от которого зависит другой атрибут, называется *детерминантом* отношения.

Для функциональных зависимостей как фундаментальной основы проекта БД были проведены исследования, позволяющие избежать избыточного их представления

Ряд зависимостей могут быть выведены из других путем применения правил, названных аксиомами Армстронга

-*Рефлексивность:* если B является подмножеством A, то A->B

-*Дополнение:* если. A->B , то AC->BC

-*Транзитивность:* если A->B и B->C , то A->C.

Отношение находится в первой нормальной форме тогда и только тогда, когда на пересечении каждого столбца и каждой строки находятся только элементарные значения атрибутов.

Отношения, находящиеся в первой нормальной форме, часто называют просто нормализованными отношениями. Соответственно, ненормализованные отношения могут интерпретироваться как таблицы с неравномерным заполнением

Препода-ватель	**День недели**	**Номер пары**	**Название дисциплин**	**Тип занятий**	**Группа**
Петров В. И.	Понед.	1	Теор. выч. проц.	Лекция	4906
	Вторник	1	Комп. графика	Лаб. раб.	4907
	Вторник	2	Комп. графика	Лаб. раб.	4906
Киров В. А.	Понед.	2	Теор. иформ.	Лекция	4906
	Вторник	3	Упр.дан.	Лаб. раб.	4907

Отношение находится во второй нормальной форме тогда и только тогда, когда оно находится в первой нормальной форме и не содержит неполных функциональных зависимостей непервичных атрибутов от атрибутов первичного ключа.

Рассмотрим отношение, моделирующее сдачу студентами текущей сессии. Структура этого отношения определяется следующим набором атрибутов:

(ФИО. Номер зач.кн.. Группа. Дисциплина. Оценка)

Для приведения данного отношения ко второй нормальной форме следует разбить его на проекции, при этом должно быть соблюдено условие восстановления исходного отношения без потерь. Такими проекциями могут быть два отношения:

(ФИО, Номер.зач.кн.. Группа) (Номер зач.кн.. Дисциплина. Оценка)

Этот набор отношений не содержит неполных функциональных зависимостей, и поэтому эти отношения находятся во второй нормальной форме

Отношение находится в третьей нормальной форме тогда и только тогда, когда оно находится во второй нормальной форме и не содержит транзитивных зависимостей.

(ФИО. Номер зач.кн.. Группа. Факультет, Специальность, Выпускающая кафедра)

Первичным ключом отношения является Номер зач.кн., однако рассмотрим остальные функциональные зависимости

- Номер зач .кн. -> ФИО
- Номер зач.кн. -> Группа
- Номер зач.кн. -> Факультет
- Номер зач.кн. -> Специальность
- Номер зач.кн. -> Выпускающая кафедра
- Группа -> Факультет
- Группа -> Специальность
- Группа -> Выпускающая кафедра
- Выпускающая кафедра -> Факультет

Эти зависимости образуют транзитивные группы

Для того чтобы избежать этого, мы можем предложить следующий набор отношений: (Номер. зач. кн., ФИО. Специальность. Группа) (Группа. Выпускающая кафедра) (Выпускающая кафедра, Факультет)
Полученный набор отношений находится в третьей нормальной форме.
Отношение находится в нормальной форме Бойса—Кодда, если оно находится в третьей нормальной форме и каждый детерминант отношения является возможным ключом отношения.
Отношение, которое моделирует сдачу текущей сессии, имеет следующую структуру: (Номер зач.кн.. Идентификатор_студента. Дисциплина. Дата. Оценка)
Возможными ключами отношения являются :Нонер_зач.кн, Дисциплина, Дата и Идентификатор_студента, Дисциплина, Дата.
Какие функциональные зависимости у нас имеются?
Номер_зач.кн, Дисциплина. Дата -> Оценка;
Идентификатор_студента, Дисциплина. Дата -> Оценка;
Номер зач.кн. -> Идентификатор_студента;
Идентификатор_студента -> Номер зач.кн.

Для приведения отношения к нормальной форме Бойса—Кодда надо разделить отношение, например, на два со следующими схемами:
(Идентификатор_студента. Дисциплина. Дата. Оценка)

(Номер зач.кн.. Идентификатор_студента) или наоборот:

(Номер зач.кн., Дисциплина. Дата, Оценка)

(Номер зач.кн.. Идентификатор_студента)
В большинстве случаев достижение третьей нормальной формы или даже формы Бойса—Кодда считается достаточным для реальных проектов баз данных, однако в теории нормализации существуют нормальные формы высших порядков, которые уже связаны не с функциональными зависимостями между атрибутами отношений, а отражают более тонкие вопросы семантики предметной области и связаны с другими видами зависимостей.

В отношении R (A, B, C) существует многозначная зависимость (multi valid dependence, MVD) R.A -» R.B в том и только в том случае, если множество значений B, соответствующее паре значений A и C, зависит только от A и не зависит от C.

При рассмотрении многозначных зависимостей мы выделяем случаи, когда одному значению некоторого атрибута соответствует устойчиво постоянное множество значений другого атрибута

Дальнейшая нормализация отношений, подобных нашему, основывается на теореме Фейджина.

Отношение R (A, B, C) можно спроецировать без потерь в отношения R1 (A, B) и R2 (A, C) в том и только в том случае, когда существует MVD A -» B C (что равнозначно наличию двух зависимостей A -» B и A -» C).

Под проецированием без потерь понимается такой способ декомпозиции отношения путем применения операции проекции, при котором исходное отношение полностью и без избыточности восстанавливается путем естественного соединения полученных отношений. Практически теорема доказывает наличие эквивалентной схемы для отношения, в котором существует несколько многозначных зависимостей

Отношение R находится в четвертой нормальной форме (4NF) в том и только в том случае, если в случае существования многозначной зависимости A → B все остальные атрибуты R функционально зависят от A.

В нашем примере можно произвести декомпозицию исходного отношения в два отношения:

(Номер зач.кн. Группа)

(Группа, Дисциплина)

Оба эти отношения находятся в 4NF и свободны от отмеченных аномалий

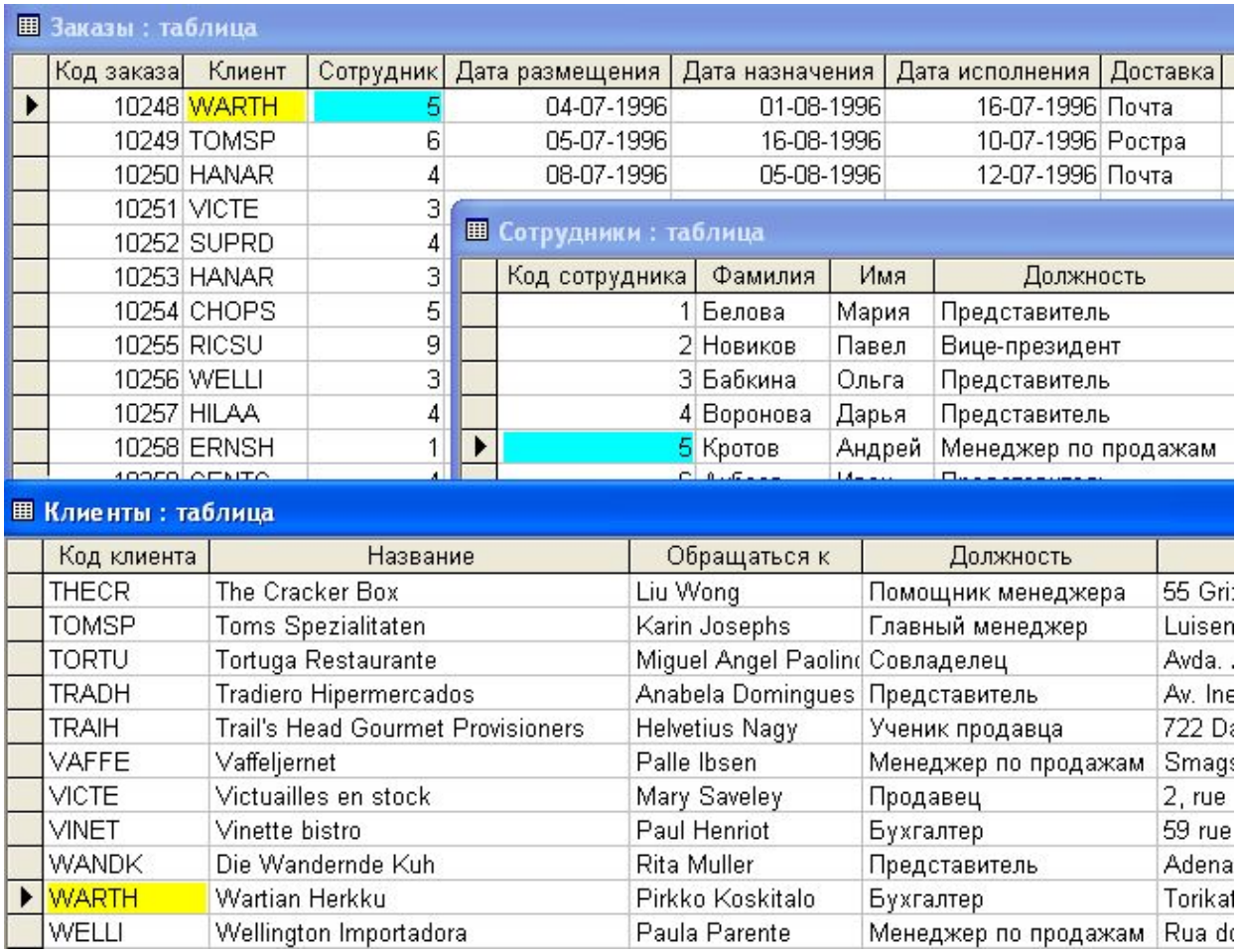
Последней нормальной формой является пятая нормальная форма 5NF, которая связана с анализом нового вида зависимостей, *зависимостей «проекции соединения» (project-join зависимости,* обозначаемые как *PJ-зависимости).* Этот вид зависимостей является в некотором роде обобщением многозначных зависимостей.

Отношение R (X, Y, ..., Z) удовлетворяет зависимости соединения (X, Y, ..., Z) в том и только в том случае, когда R восстанавливается без потерь путем соединения своих проекций на X, Y, ..., Z. Здесь X, Y, ..., Z — наборы атрибутов отношения R.

Отношение R находится в пятой нормальной форме (нормальной форме проекции-соединения — PJ/NF) в том и только в том случае, когда любая зависимость соединения в R следует из существования некоторого возможного ключа в R.

Пятая нормальная форма редко используется на практике. В большей степени она является теоретическим исследованием.

Взаимосвязь таблиц БД



Заказы : таблица

Код заказа	Клиент	Сотрудник	Дата размещения	Дата назначения	Дата исполнения	Доставка
10248	WARTH	5	04-07-1996	01-08-1996	16-07-1996	Почта
10249	TOMSP	6	05-07-1996	16-08-1996	10-07-1996	Ростра
10250	HANAR	4	08-07-1996	05-08-1996	12-07-1996	Почта
10251	VICTE	3				
10252	SUPRD	4				
10253	HANAR	3				
10254	CHOPS	5				
10255	RICSU	9				
10256	WELLI	3				
10257	HILAA	4				
10258	ERNSH	1				

Сотрудники : таблица

Код сотрудника	Фамилия	Имя	Должность
1	Белова	Мария	Представитель
2	Новиков	Павел	Вице-президент
3	Бабкина	Ольга	Представитель
4	Воронова	Дарья	Представитель
5	Кротов	Андрей	Менеджер по продажам

Клиенты : таблица

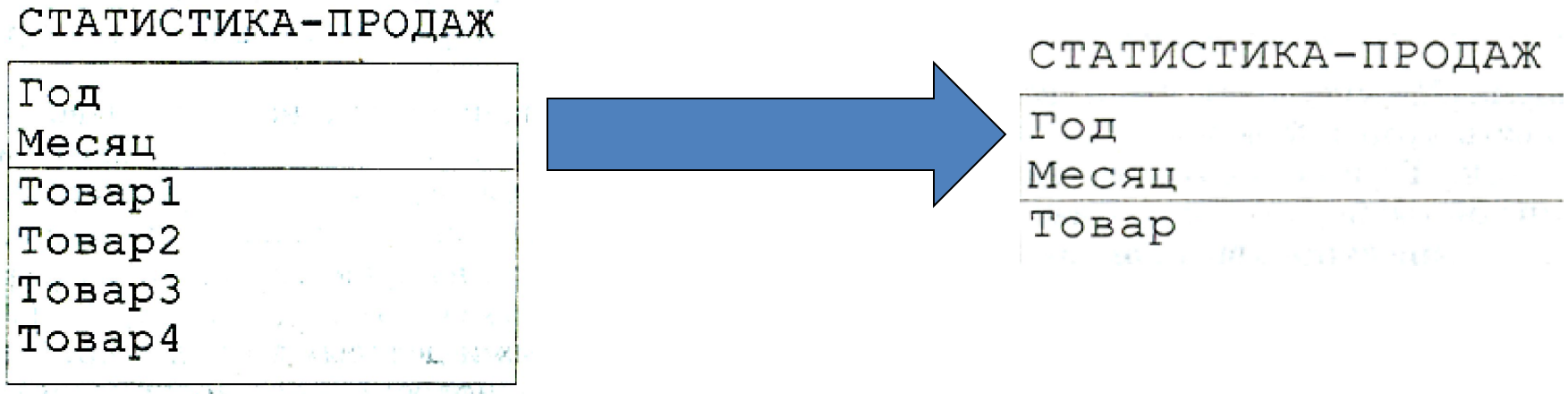
Код клиента	Название	Обращаться к	Должность	
THECR	The Cracker Box	Liu Wong	Помощник менеджера	55 Gri
TOMSP	Toms Spezialitaten	Karin Josephs	Главный менеджер	Luisen
TORTU	Tortuga Restaurante	Miguel Angel Paolin	Совладелец	Avda. .
TRADH	Tradiero Hipermercados	Anabela Domingues	Представитель	Av. Ine
TRAIH	Trail's Head Gourmet Provisioners	Helvetius Nagy	Ученик продавца	722 Da
VAFFE	Vaffeljernet	Palle Ibsen	Менеджер по продажам	Smags
VICTE	Victuailles en stock	Mary Saveley	Продавец	2, rue
VINET	Vinette bistro	Paul Henriot	Бухгалтер	59 rue
WANDK	Die Wandernde Kuh	Rita Muller	Представитель	Adena
WARTH	Wartian Herkku	Pirkko Koskitalo	Бухгалтер	Torikat
WELLI	Wellington Importadora	Paula Parente	Менеджер по продажам	Rua do

Нормализация таблиц

Цель - устранение избыточности данных

1-я нормальная форма (1НФ) – каждое поле:

- должно быть неделимым;
- не должно содержать повторяющихся групп.



2НФ - все поля должны зависеть от первичного ключа, то есть чтобы первичный ключ однозначно определял запись

Пример нормализации

Накладная № 123

Дата	Покупатель	Адрес
10.01.97	ТОО "Геракл"	г. Москва, ул. Стромынка, 20

Отпущен товар	Количество	ед. изм.	Цена ед.изм.	Общая стоимость
Тушенка	10000	банки	7000	70 000 000
Сахар	200	кг	5000	1 000 000
Макароны	1000	кг	3000	3 000 000

Итого 74 000 000

1 НФ

ОТПУСК-ТОВАРОВ-СО-СКЛАДА

Дата
Покупатель
Город
Адрес
Товар
Ед_измерения
Цена_за_ед_изм
Отпущено_ед
Общая_стоимость
Номер_накладной

2 НФ

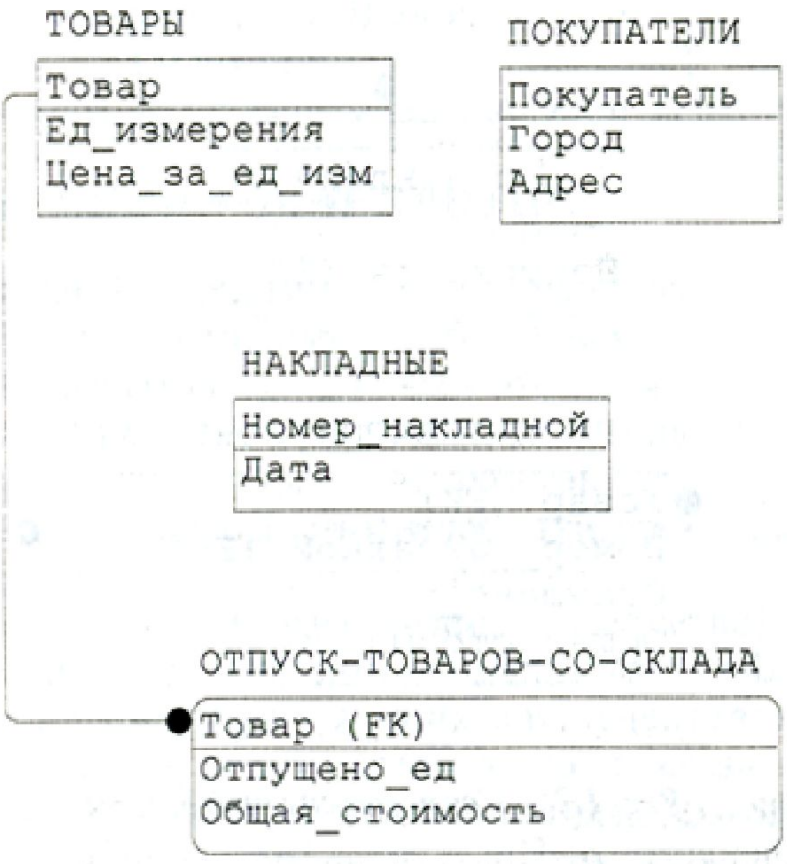
ОТПУСК-ТОВАРОВ-СО-СКЛАДА

Номер_накладной
Товар

Дата
Покупатель
Город
Адрес
Ед_измерения
Цена_за_ед_изм
Отпущено_ед
Общая_стоимость

Пример нормализации. Окончательное приведение ко 2 НФ.

Установка связей.



Пример нормализации. 3 НФ

3НФ - значение любого поля, не входящего в первичный ключ, не должно зависеть от значения другого поля, не входящего в первичный ключ