

Дисциплина: Базы данных



«Нормальные формы, ER-диаграммы»

**Преподаватель:
Лут А.В.**

Нормальные формы

Понятие нормальных форм позволяет судить о том, хороша или плоха схема базы данных, и если плоха, то почему и дает алгоритмы исправления ситуации.

Тот факт, что схема базы данных имеет недостатки, зачастую порождается **избыточностью хранения информации**.

Нормализация отношений преследует **простую цель**: *каждый факт в базе данных должен храниться в единственном экземпляре*.

Первая нормальная форма

Определение. Отношение находится в *первой нормальной форме*, если оно не содержит повторяющихся групп.

Приведем пример отношения, нарушающего первую нормальную форму

Табел ьный номер	Фамилия	Дата приема на работу	Ребенок 1		...	Ребенок N	
			Имя	Дата рождения	...	Имя	Дата рождения

Первая нормальная форма нарушена, так как отношение содержит повторяющуюся неопределенное число раз группу «Ребенок».

Приведение к первой нормальной форме можно выполнить, если разбить это отношение на два:

- 1) сотрудник (Табельный номер, Фамилия, Дата приема на работу);
- 2) дети (Табельный номер, Имя ребенка, Дата рождения).

Большинство СУБД не позволяют использовать отношения, нарушающие первую нормальную форму.

Декомпозиции отношений

Замена отношения на два или более других называется *декомпозицией*. Например, отношение $R(A, B, C)$ можно попытаться заменить на два отношения $R1(A, B)$ и $R2(A, C)$. Устранение избыточности выполняется именно как декомпозиция.

Выясним какими свойствами должна обладать декомпозиция, чтобы иметь право на существование.

1-е свойство: соединения без потерь информации

Говорят, что декомпозиция отношения R на R1, R2 обладает *свойством соединения без потерь*, если соединение R1, R2 дает исходное отношение R:

$$R=R1*R2.$$

Рассмотрим отношение R (Фамилия, Рост, Вес). Будем считать, что все фамилии уникальны, то есть, поле «**Фамилия**» является первичным ключом. Ниже представлен экземпляр такого отношения.

Фамилия	Рост	Вес
Иванов	180	75
Петров	180	85
Сидоров	175	73

Декомпозиция его на R1(Фамилия, Рост) и R2(Фамилия, Вес) дает отношения:

Фамилия	Рост
Иванов	180
Петров	180
Сидоров	175

Фамилия	Вес
Иванов	75
Петров	85
Сидоров	73

Легко убедиться, что соединение R1, R2 по полю фамилия даст исходное отношение R.

Фамилия	Рост	Вес
Иванов	180	75
Петров	180	85
Сидоров	175	73

Рассмотрим теперь декомпозицию R на R1 (Фамилия, Рост) и R2 (Рост, Вес).

Фамилия	Рост
Иванов	180
Петров	180
Сидоров	175

Рост	Вес
180	75
180	85
175	73

Соединение этих отношений по полю «Рост» дает результат:

Фамилия	Рост	Вес
Иванов	180	75
Иванов	180	85
Петров	180	75
Петров	180	85
Сидоров	175	73

Исходное отношение восстановить не удалось. Следовательно, декомпозиция не обладает свойством соединения без потерь информации и права на существование не имеет.

Наличие свойства соединения без потерь является обязательным.

Теорема. Декомпозиция отношения R на R_1, R_2 обладает свойством соединения без потерь тогда и только тогда, когда $R_1 \cap R_2 \rightarrow R_1 - R_2$ или $R_1 \cap R_2 \rightarrow R_2 - R_1$.

2-е свойство: сохранения зависимостей

Другое важное свойство декомпозиции $r = (R_1, R_2, \dots, R_N)$ схемы отношения R заключается в том, чтобы множество зависимостей F для R было выводимым из проекций F на схемы отношений R_i .

Формально проекцией F на множество атрибутов Z , обозначаемой как $\pi_Z(F)$, называется множество зависимостей $X \rightarrow Y$ в F , таких, что $XY \subseteq Z$.

Заметим, что зависимость $X \rightarrow Y$ не обязательно принадлежит F , она должна принадлежать F^+ .

Будем говорить, что декомпозиция r сохраняет зависимости F , если из объединения всех проекций $\pi_{R_i}(F)$ выводимы все зависимости из F .

Стремление к тому, чтобы декомпозиция обладала свойством сохранения зависимостей естественно. Зависимости в схеме БД могут рассматриваться как ограничения целостности и несохранение их позволит создать такие текущие значения отношений, которые не удовлетворяют F .

Пример. Пусть на множестве атрибутов
(C-город, Z-индекс, S-адрес)

определены зависимости: $Z \rightarrow C$, $CS \rightarrow Z$.

Декомпозиция на пару отношений $R_1(C, Z)$ и $R_2(S, Z)$ обладает свойством соединения без потерь (поскольку $CZ \cap SZ \rightarrow CZ - SZ$). Однако зависимость $CS \rightarrow Z$ не сохраняется, так как не следует из проекции исходного множества зависимостей на схемы отношений декомпозиции.

Кроме того, декомпозиция может сохранять F , не обладая при этом свойством соединения без потерь. Например, это имеет место для

$$F = \{A \rightarrow B, C \rightarrow D\}, R = (ABCD),$$

и декомпозиции $r = \{AB, CD\}$.

Вторая нормальная форма

Определение. Атрибут или группа атрибутов А частично зависит от группы атрибутов В, если существует функциональная зависимость $X \rightarrow A$, где $X \subseteq В$.

Пример

№ проекта	Название проекта	№ этапа	Дата начала этапа	Дата окончания этапа
1	Разработка базы данных	1	01.01.2024	31.07.2024
1	Разработка базы данных	2	01.08.2024	31.12.2024
2	Текстовый редактор	1	11.11.2024	31.08.2024

В этом примере «**Название проекта**» упоминается много раз вместе с каждым этапом этого проекта. Первичным ключом отношения является «№ проекта», «№ этапа». В то же время, существует функциональная зависимость

«№ проекта» → «Название проекта»,

что означает, что атрибут «**Название проекта**» частично зависит от ключа, так как атрибут «**№ проекта**» является частью ключа. Отношение содержит избыточную информацию, которая порождает так называемые **аномалии**.

Аномалия вставки

При добавлении нового этапа в проект, необходимо вновь указать название проекта, которое может быть введено не совпадающим с названием проекта уже имеющемся в описании других этапов. В результате отношение будет содержать противоречивые данные.

Кроме того, когда появляется новый проект, для которого еще не выполнена разбивка на этапы, некуда поместить информацию о номере и названии проекта. Если ее все же занести в отношение, то поля, относящиеся к этапу, останутся пустыми и оператор может забыть, что позднее он должен их заполнить.

Аномалия модификации

При изменении названия проекта, изменённое название необходимо внести во все этапы проекта, в противном случае отношение будет содержать внутренние противоречия.

Аномалия удаления

При удалении всех этапов проекта исчезает вся информация о проекте.

Отношение из приведенного примера можно нормализовать, разбив его на два: **Проект** (№ проекта, Название проекта); **Этап** (№ проекта, № этапа, Дата начала этапа, Дата окончания этапа).

Отношения примут вид:

Отношение **Проект**

№ проекта	Название проекта
1	Разработка базы данных
2	Текстовый редактор

Отношение **Этап**

№ проекта	№ этапа	Дата начала этапа	Дата окончания этапа
1	1	01.01.2024	31.07.2024
1	2	01.08.2024	31.12.2024
2	1	11.11.2024	31.08.2024

Полученная декомпозиция лишена недостатков исходного отношения.

Определение. Отношение находится во второй нормальной форме, если каждый его непервичный атрибут полностью (не частично) зависит от первичного ключа.

Пример. Пусть на отношении $R(a, b, c, d)$ определены зависимости:
$$a, b \rightarrow c; b \rightarrow d.$$

Отношение нарушает вторую нормальную форму, так как непервичный атрибут d частично зависит от ключа a, b (вторая зависимость).

Нормализация заключается в расщеплении отношения на два:
 $R1(a, b, c)$ и $R2(b, d)$.

Нетрудно убедиться, что декомпозиция обладает свойством соединения без потерь информации.

Третья нормальная форма

Пусть дано отношение $R(A, B, C)$ и на нем определены зависимости:

$$A \rightarrow B \quad B \rightarrow C,$$

но зависимость $B \rightarrow A$ не имеет места. В этом случае говорят, что A транзитивно определяет C . Действительно, зависимость $A \rightarrow C$ может быть выведена из исходных зависимостей, так как по аксиоме транзитивности

из $A \rightarrow B$ и $B \rightarrow C$, следует $A \rightarrow C$.

Определение. Отношение находится в третьей нормальной форме, если каждый первичный атрибут нетранзитивно зависит от первичного ключа.

Пример. Пусть дано отношение

$R(\text{Магазин, Отдел, Товар, Зав. отделом}).$

Отношение описывает ряд магазинов, которые делятся на отделы; в каждом отделе имеется заведующий отделом и в этих отделах продаются товары. На отношении определены функциональные зависимости:

Магазин, Отдел \rightarrow Зав. отделом – отдел имеет одного заведующего;

Магазин, Товар \rightarrow Отдел – каждый товар в магазине продается в единственном отделе.

Отношение R находится во второй нормальной форме, так как его первичный ключ **Магазин, Товар** полностью определяет непервичные атрибуты

Отдел, Зав. отделом.

Однако, отношение избыточно, что хорошо видно из примера экземпляра этого отношения.

Магазин	Отдел	Зав. отделом	Товар
М-Видео	Радиотовары	Иванов	Телевизор
М-Видео	Радиотовары	Иванов	MP3-player
М-Видео	Радиотовары	Иванов	Радиоприемник
М-Видео	Радиотовары	Иванов	Антенна
...

Фамилия зав. отделом **Иванов** повторяется многократно вместе с каждым товаром. Отношение избыточно и это порождает все те же **аномалии**.

Аномалия вставки.

До тех пор, пока в отделе не продается хотя бы один товар, некуда поместить фамилию заведующего.

Аномалия модификации.

При изменении поля **«Зав. отделом»** необходимо модифицировать множество кортежей. Существует возможность того, что не все кортежи будут модифицированы, в результате чего отношение будет противоречивым.

Аномалия удаления.

При удалении всех товаров, продаваемых в отделе, исчезает информация о заведующем.

В данном случае избыточность порождена нарушением третьей нормальной формы. Действительно:

Магазин, Товар → Магазин, Отдел → Зав. отделом,
и зависимость

Магазин, Отдел → Магазин, Товар
не имеет места.

Следовательно, атрибуты **Магазин, Товар** транзитивно определяют первичный атрибут **Зав. отделом**.

Избыточность может быть устранена расщеплением отношения на два:

Отдел (Магазин, Отдел, Зав. отделом)

Товар (Магазин, Товар, Отдел)

Магазин	Отдел	Зав. отделом	Отношение
М-Видео	Радиотовары	Иванов	
...	

Отношение Отдел	Магазин	Отдел	Товар
	М-Видео	Радиотовары	Телевизор
	М-Видео	Радиотовары	MP3-player
	М-Видео	Радиотовары	Радиоприемник
	М-Видео	Радиотовары	Антенна

Теперь тот факт, что Иванов заведует отделом Радиотовары в магазине М-Видео, хранится в базе данных в единственном экземпляре.

Все отношения базы данных должны находиться в третьей нормальной форме. Как правило, не существует мотивов, которые бы оправдывали нарушение третьей нормальной формы.

Нормальная форма Бойса-Кодда

Отношение может находиться в третьей нормальной форме, но, тем не менее, быть избыточным. Рассмотрим пример:

R(Город, Адрес, Индекс)

Пример экземпляра такого отношения приведен ниже:

Город	Адрес	Индекс
Челябинск	Пр. Ленина 45	454080
Челябинск	Пр. Ленина 71	454080
Пермь	Пр. Ленина 45	626090

На отношении определены зависимости:

Индекс \rightarrow Город

Город, Адрес \rightarrow Индекс

Тот факт, что индекс 454080 относится к Челябинску, отражен в отношении неоднократно.

Определение. Детерминантой называется левая часть любой функциональной зависимости, определенной на отношении.

Отношение находится в нормальной форме Бойса – Кодда, если каждая его детерминанта является возможным первичным ключом.

В нашем примере детерминанта **Индекс** первичным ключом не является и форма Бойса – Кодда нарушена.

Приведение к форме Бойса – Кодда может быть выполнено расщеплением исходного отношения на два:

R1(Индекс, Город)

и **R2(Адрес, Индекс).**

Экземпляры отношений примера будут иметь вид:

Адрес	Индекс
Пр. Ленина 45	454080
Пр. Ленина 71	454080
Пр. Ленина 45	626090

Отношение **R2**

Город	Индекс
Челябинск	454080
Пермь	626090

Отношение **R1**

Новое представление избыточно, но обладает недостатками:

- 1) эти отношения нельзя показывать пользователю в том виде, как они хранятся в базе данных – не поймут;
- 2) зависимость **Город, Адрес → Индекс** не сохраняется и программист будет вынужден выполнять соединение отношений, чтобы проверить, что новые данные ее не нарушают.

Таким образом, устранив избыточность, мы создали дополнительные проблемы для программиста.

Если соблюдение третьей нормальной формы обязательно, то вопрос приводить ли схему к форме Бойса-Кодда, следует решать отдельно в каждом конкретном случае.

Многозначные зависимости и четвертая нормальная форма

Отношение может находиться в форме Бойса-Кодда и, тем не менее, быть избыточным. Рассмотрим пример, в котором речь идет о выполнении пассажирских авиаперевозок.

№ рейса	Дата	Пассажир	Член экипажа
285	01.09.2024	Иванов	Петров
285	01.09.2024	Сидоров	Измайлов
285	01.09.2024	Иванов	Измайлов
285	01.09.2024	Сидоров	Петров
287	03.09.2024	Петрухин	Никитин

Тот факт, что Иванов летит в качестве пассажира рейсом 285 от 01 сентября 2024 года, в отношении отражен дважды, то же можно сказать относительно других пассажиров и членов экипажа этого рейса.

В общем случае каждый пассажир может появиться в этом отношении в кортеже вместе с каждым членом экипажа. На этом отношении не определено ни одной функциональной зависимости, однако, избыточность налицо.

В приведенном примере существует две многозначных зависимости:

№ рейса, Дата \rightarrow Пассажир

№ рейса, Дата \rightarrow Член экипажа

Определение. Отношение $R(A, B, C)$ нарушает четвертую нормальную форму, если при наличии многозначных зависимостей $A \twoheadrightarrow B$ и $A \twoheadrightarrow C$, A не является первичным ключом.

Приведение к четвертой нормальной форме осуществляется декомпозицией исходного отношения на два: **R1(A, B)** и **R2(A, C)**. В нашем примере будут получены отношения: Пассажиры(№ рейса, Дата, Пассажир); Экипаж(№ рейса, Дата, Член экипажа).

№ рейса	Дата	Пассажир
285	01.09.2024	Иванов
285	01.09.2024	Сидоров
287	03.09.2024	Петрухин

Отношение **Пассажир**

№ рейса	Дата	Член экипажа
285	01.09.2024	Петров
285	01.09.2024	Измайлов
287	03.09.2024	Никитин

Отношение **Экипаж**

Теперь мы по отдельности храним списки пассажиров и списки членов экипажа. В схеме базы данных обязательно должны присутствовать отношения, сохраняющие многозначные зависимости.

Диаграммы Сущность-Связь (ER)

Метод разработки схем баз данных, основывающийся на гораздо менее формализованных положениях, носит название «Сущность-Связь».

Диаграммы «Сущность-Связь» называют ER-диаграммами, что происходит от словосочетания Entity-Relation.

Сущности, атрибуты, связи.

Сущность определяется как тип объекта.

Экземпляры сущности — экземпляры объекта данного типа, отличающиеся друг от друга и допускающие однозначную идентификацию.

Сущностями могут быть автомашины, накладные, люди, и, даже муравьи в муравейнике, если вы придумаете способ отличать их друг от друга, например, напишете номер на каждом муравье.

Атрибут есть свойство сущности. Атрибутами сущности «Человек» могут быть имя, фамилия, номер паспорта и т.п.

Связь есть отношение между двумя или более сущностями. Примеры связей:

- муравей ЖИВЕТ В муравейнике;
- человек ЕЗДИТ НА автомашине.

Здесь муравей, муравейник, человек, автомашина — сущности, а ЖИВЕТ В, ЕЗДИТ НА — связи.

Как правило, сущности — это существительные, а связи — глаголы. На диаграммах сущности изображают прямоугольниками, а связи — ромбами.

Пример:



К сожалению, приведенные определения сущности, связи и атрибута не отличаются особой конкретностью.

Атрибут или набор атрибутов, используемых для идентификации экземпляра сущности, является ключом сущности. Каждый экземпляр связи однозначно определяется набором ключей сущностей, соединяемых этой связью.

Степень связи

Рассмотрим бинарную связь <ЧИТАЕТ> между сущностями <ПРЕПОДАВАТЕЛЬ> и <КУРС>. Степень этой связи может принимать значения:

1:1 – один к одному

1:M – один ко многим

M:1 – многие к одному

M:M – многие ко многим

Еще одной характеристикой связи является ее обязательность или необязательность. Если экземпляры данной сущности могут не участвовать в связи, то связь называют необязательной. Будем обозначать обязательность связи буквами <н> и <о>.

Тогда возможны следующие характеристики связи между сущностями <ПРЕПОДАВАТЕЛЬ> и <КУРС>:

$1_n:1_n$ – каждый преподаватель читает не более, чем один курс и каждый курс читается не более, чем одним преподавателем. При этом преподаватель может не читать ни одного курса, а некоторые курсы могут не читаться вовсе.

$1_0:1_n$ – каждый преподаватель читает ровно 1 курс, но курс не обязательно кем то читается.

$1_n:1_0$ – преподаватель читает не более чем один курс, но может не читать ничего. Каждый курс читает ровно один преподаватель.

$1_0:1_0$ – каждый преподаватель обязательно читает ровно один курс, и каждый курс читается одним преподавателем.

$1_n:M_n$ – преподаватель читает 0, 1, ..., n курсов, каждый курс читает не более чем 1 преподаватель. Курсы могут не читаться вовсе.

Можете продолжить толкование комбинаций самостоятельно.

Построение отношений на основании ER-диаграмм.

Фрагмент ER–диаграммы, включающий две сущности и одну связь может быть преобразован в одно, два или три отношения в зависимости степени и определённости связи.

Степень 1:1

Если степень бинарной связи равна 1:1 и класс принадлежности обеих сущностей связи является обязательным, то требуется для отображения сущностей и связи на схему БД только одно отношение. Первичными ключами этого отношения являются ключи каждой из сущностей. Для приведенного примера **ПРЕПОДАВАТЕЛЬ ЧИТАЕТ КУРС**, отношение имеет вид:

$$\mathbf{R11 = (нп, вап, нк, вак)}$$

Обозначения: вап – вторичные атрибуты преподавателя; нк – номер курса; вак – вторичные атрибуты курса; нп – номер преподавателя.

Если для одной или обеих сущностей класс принадлежности связи является необязательным, то, если мы сохраним отношение в прежнем виде, то допустим избыточность.

Действительно, пусть существуют преподаватели, которые ничего не читают, тогда в отношении R11 существуют кортежи, в которых данные курса являются неопределенными.

Избыточность можно устранить, разбив отношение на два:

$$\mathbf{R21=(np, вап); \quad R22=(np, нк, вак)}$$

Теперь в отношении R22 имеются кортежи только для тех преподавателей, которые что то читают.

При этом существует ограничение целостности БД, которое заключается в том, что каждой записи из R21 может соответствовать не более чем одна запись в R22.

Если же для обеих сущностей принадлежность к связи является необязательной, то требуется уже три отношения:

$$R31=(np, vap); \quad R32=(nk, vak); \quad R33=(np, nk).$$

Степень 1:М

Рассмотрим случай, когда преподаватель может читать много курсов, но каждый курс читается не более, чем одним преподавателем. Выбор решения зависит лишь от того, является ли принадлежность связи для курса обязательной. Если да, то требуется два отношения: $R41=(np, vap); \quad R42=(np, nk, vak).$

В противном случае потребуются три отношения:

$$R51=(np, vap); \quad R52=(nk, vak); \quad R53=(np, nk),$$

при этом существует ограничение целостности, заключающееся в том, что каждому курсу может соответствовать не более чем один преподаватель.

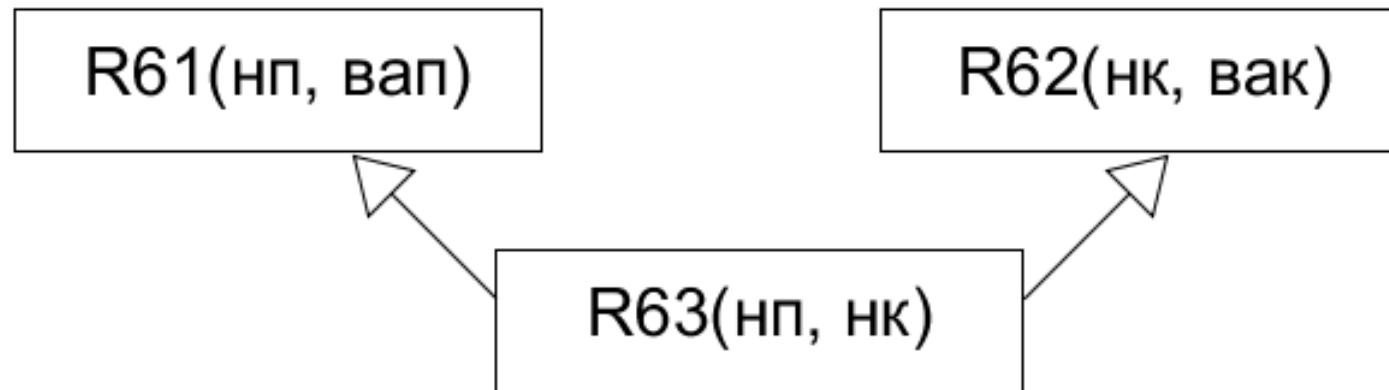
Степень М:М

Для рассмотренного выше примера Преподаватель—Курс связь имеет степень М:М, когда преподаватель может читать много курсов и каждый курс может читаться более чем одним преподавателем.

Вне зависимости от обязательности принадлежности сущности связи, требуется три отношения:

R61=(нп, вап); R62=(нк, вак); R63=(нп, нк).

Дополнительные ограничения целостности в этом случае отсутствуют. Схема соответствующего фрагмента базы данных имеет вид:



Отношение R63 связывает связью М:М(многие ко многим) отношение R61(преподаватель) и отношение R62(курс).

Связи более высокого порядка.

Рассмотренные случаи касались бинарных связей, то есть связей между двумя сущностями.

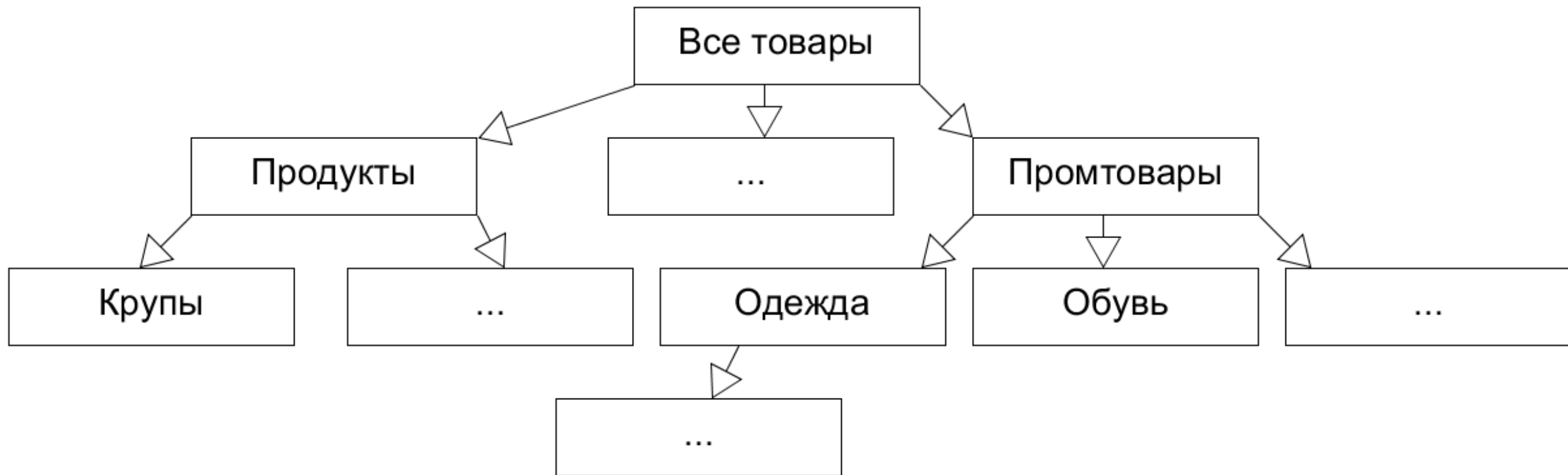
Часто приходится иметь дело со связями более высокого порядка, которые не могут быть сведены к набору бинарных связей.

Примером может служить расписание занятий в университете, которое является набором кортежей вида:

(аудитория, группа, преподаватель, курс, время занятий).
Здесь аудитория, группа, преподаватель, курс – ключи сущностей, а время занятий – вторичный атрибут связи.

Деревья и простые сети в схемах баз данных

Примером дерева в базе данных может служить любая классификация, например, классификация товаров:



Любая такая классификация может быть построена на единственном отношении со схемой:

**(собственный ключ, ключ родительской записи,
вторичные атрибуты)**

Для приведенного примера множество записей имеет вид:

Собственный ключ	Родительский ключ
Все товары	
Продукты	Все товары
Промтовары	Все товары
Крупы	Продукты
Одежда	Промтовары
Обувь	Промтовары
...	...

Такое дерево на схеме базы данных имеет вид петли:



Простая сеть – граф, не содержащий циклов. Примером может служить генеалогическое дерево, которое также строится на одном отношении со схемой:

**(собственный ключ, ключ отца,
ключ матери, вторичные атрибуты)**

На схеме базы данных это будет иметь вид:



Соб. имя	Имя отца	Имя матери
Адам		
Ева		
Авель	Адам	Ева
Каин	Адам	Ева
Енох	Каин	?
...

Домашнее задание (прочитать и ответить на вопросы).

1. «зависимости соединения» (стр. 18-19).
2. Определите понятия «сущность», «атрибут», «связь».
3. Что такое степень связи и обязательность связи.
4. Какие схемы баз данных порождаются при различных сочетаниях степени и обязательности связи.
5. Отображение деревьев, множеств и простых сетей на схемы баз данных.