

Вехова Д.В.,

Студент,

2 курс, факультет математики и информационных технологий,

Башкирский Государственный Университет,

Россия, г. Стерлитамак

Хусаинова Г.Я., к. ф.-м. н., доцент,

доцент кафедры прикладной информатики и программирования,

Башкирский Государственный Университет

Россия, г. Стерлитамак

НОРМАЛИЗАЦИЯ БАЗ ДАННЫХ

***Аннотация:** В статье рассматривается тема нормализации баз данных. Даны определения самой нормализации и ее видам. Зачем нужна нормализация ее плюсы и минусы. Так же какой вид нормализации считается самым оптимальным и как его достичь.*

***Ключевые слова:** База данных, нормализация, 1NF, 2NF, 3NF, BCNF, 4NF, 5NF, DKNF, 6NF.*

***Annotation:** The article deals with the topic of database normalization. The definitions of the normalization and its kinds are given. Why do we need to normalize its pros and cons. Also what kind of normalization is considered the most optimal and how to achieve it.*

***Key words:** Database, normalization, 1NF, 2NF, 3NF, BCNF, 4NF, 5NF, DKNF, 6NF.*

Нормализация. В работе с базами данных мы встречаемся с такими проблемами как повторяющиеся группы и другие различные противоречия которые не позволяют корректное изменение данных. Нормализация – это явление, в реляционной базе данных, которое было разработано, чтобы уменьшить избыточность данных [1, ст. 18].

Когда мы используем ненормализованные таблицы, то очень большая вероятность, что целостность данных будет нарушена. Когда таблицы ненормализованные, то возникают такие проблемы как:

1. избыточность данных (одна и та же информация в нескольких таблицах);
2. аномалия обновления (необходимо вводить исправления в данных во все таблицы, они используются);
3. аномалия удаления (возможна потеря нужной информации при удалении);
4. аномалия ввода (если для каких-то полей базы данных заданы ограничения и невозможно сразу ввести данные).

Разберем аномалии на примерах. Избыточность: предположим у нас есть компания, у этой компании есть адрес, телефон и различные данные, так же в ней работают сотрудники, когда их небольшое количество данные о компании вводить ещё нормально, но когда их много, то это становится сложно и занимает значительно больше времени. Но если бы мы могли ввести данные о компании в одну таблицу, и они запоминались автоматически при добавлении новых сотрудников. Тогда не придется писать все её данные, а нужно будет только выбрать нужная компания. Обновление: так же опираясь на нашу компанию, предположим, что по каким-то причинам она сменила адрес, телефон или еще какие-то данные. Из-за этого нам придется вводить новые данные во все таблицы, где они находятся, согласитесь это не очень удобно, гораздо проще изменить данные только в одном месте и переприсвоить значения во все остальные. Удаление: рассмотрим такую ситуацию, в компании работал электрик (они работал только один, больше в компании электриков не было), но потом он решил уволиться, то есть из базы данных удалится информация не только о данных этого сотрудника (фамилия, имя, отчество, адрес), но и полностью о его должности, заработной плате и обязанностях. Это произойдет потому что

больше сотрудников с такой должностью пока что нет и когда компания наймет нового электрика их придётся вводить заново. Ввод: аномалия ввода возникает, когда из-за ограничения на ввод нельзя сразу ввести информацию.

Нормальные формы. Как раз чтобы таких проблем не было и нужно применить процесс нормализации. Нормализацией считается метод исследования отношений на базе их существующих функциональных зависимостей и первичных и потенциальных ключей. В нем имеется ряд различных правил, использующиеся для проверки всех отношений базы данных. Известны следующие нормальные формы: первая нормальная форма, вторая нормальная форма, третья нормальная форма, нормальная форма Бойса — Кодда, четвертая нормальная форма, пятая нормальная форма, доменно-ключевая нормальная форма, шестая нормальная форма[4]. В данной статье мы разберём начальные шесть видов нормальных форм.

Нормированные формы обязаны выполнять следующее:

- Каждая нормальная форма бесспорно мощнее прошлой формы;
- Свойства предыдущих нормальных форм остаются и для следующих.

Первая нормальная форма. Считается, что если все атрибуты отношения имеют атомарные значения, то оно находится в 1НФ.

Первая нормальная форма значит образцом для всех реляционных моделей, отсюда можно сделать вывод, они уже находятся в 1НФ [3].

Вторая нормальная форма. Если использовать аппарат функциональной зависимости, то мы можем таблицы переделать под нормальную форму выше первой.

Частичная функциональная зависимость называется нечто иное как связанность неключевого атрибута с частью составного ключа.

Зависимость неключевого атрибута от всего составного ключа называется полной функциональной зависимостью.

Если отношение находится в 1НФ и каждый неключевой атрибут полностью зависит от первичного ключа, то отношение находится во второй нормальной форме [1, ст. 43].

Третья нормальная форма. Отношение находится в 3НФ, если оно находится во второй нормальной форме и все неключевые атрибуты нетранзитивно зависят от первичного ключа.

Нормальная форма Бойса-Кодда. Основную массу баз данных доводят только до 3НФ. Это считается довольно удовлетворительно. Но при использовании 3НФ не учитываются несколько важных моментов, такие как:

1. количество потенциальных ключей может быть более одного;
2. возможность потенциального ключа быть сложным;
3. что потенциальные ключи могут перекрывают друг друга.

Рассмотрим данные моменты на примере такой таблицы с данными как «школьник – учитель – предмет»[2, ст. 26]. Необходимо для примера ввести несколько правил: один учитель ведет только один предмет, один ученик может слушать данный предмет только у одного учителя, один и тот же предмет могут преподавать несколько учителей.

Таблица 1.

«Школьник – учитель – предмет»

Школьник	Предмет	Учитель
Иванов И.Ф.	Алгебра	Мишин С.К.
Иванов И.Ф.	История	Наливайка У.Б.
Богословский Н.П.	Алгебра	Ахметов У.Е.
Богословский Н.П.	История	Держинский О.Б.
Мишин Я.Б.	История	Держинский О.Б.

Из данной таблицы можно сделать вывод что она имеет избыточность, так же присутствуют возможные аномалии удаления, и аномалия изменение тоже тут есть. Данная таблица имеет несколько потенциальных ключей, такие как {Школьник, Предмет}, {Школьник, Учитель}. Данные ключи являются сложными и они перекрываются. Чтобы разрешить данные проблемы и привести таблицу в НФБК нужно разбить одну таблицу на две например на школьник и предмет в одной учитель в другой (у школьника есть какой-то предмет, этот предмет ведёт учитель) или учителя и школьник в вместе а предмет в другой (школьник ходит на уроки к такому то учителя на такой то предмет).

Переменная отношения находится в НФБК, когда каждые её неприводимые и нетривиальные слева функциональные зависимости имеют какой-либо потенциальный ключ вместо своего детерминанта.

Четвертая нормальная форма. С помощью НФБК устраняются любые аномалии обновления, если они вызваны функциональными зависимостями. Но в последующем изучении баз данных найден такой тип зависимостей как многозначная зависимость.

Переменная отношения стоит в четвёртой нормальной форме, когда она не имеет нетривиальных многозначных зависимостей и стоит в НФБК.

Перед нами встает проблема многозначных зависимостей. Она возникает потому из-за многозначных атрибутов. Решить проблему можно, если определить каждый многозначный атрибут в собственную таблицу. Но нужно не забыть определить в эту таблицу и ключ, от которого зависит атрибут.

Пятая нормальная форма. Результатом нормализации всех предыдущих схем отношений были два новых отношения. К сожалению, поделить отношение на два других может и не получиться. Тогда делается декомпозиция исходного отношения, на отношения, число которых больше двух.

Отношение R находится в 5НФ в том и только том случае, если в нем не существует зависимых соединений.

В статье были разобраны несколько начальных видов нормализации, ещё раз повторяюсь, что в основном нормализацию завершают по достижению 3НФ. Так как, начиная с НФБК, уже идёт углубленная нормализация, которая ведет к тому, что запросы будут уже стоиться сложнее, и так же они будут отнимать больше времени на их создание. Поэтому приходится выбирать, что вы хотите больше сохранить, свое время или нервы, ведь порой сконструировать нужные запросы получается не сразу. И чтобы привести к формам высших порядков, необходимо затратить довольно много времени на разбиение таблиц. Так же это приводит к тому, что запросы становятся более сложными. На это можно посмотреть и с другой стороны, вы будите более тщательнее конструировать запросы, чтобы не тратить время на повторные попытки и навыки с запросами будут только увеличиваться. Поэтому приходится делать денормализацию данных в таблицах, зато в этом есть и свой плюс так как, это приводит к тому, что запросы будет легче сконструировать и скорость их обработки будет значительно меньше, чем при нормальных формах высших порядков.

Использованные источники:

1. Избачков Ю., Петров В. Информационные системы: Учебник для вузов.– 2-е изд. – СПб.: Питер, 2014. – 656 с.
2. Хусаинова Г.Я.,Хусаинов И.Г. Информационные системы. Управление реляционными базами данных: Учеб.-метод. материалы: Стерлитамак: Стерлитамак, гос. пед. академия, 2007. – 55 с.
3. Хусаинова Г.Я., Хусаинов И.Г. Разработка автоматизированного рабочего места менеджера по продаже автомобилей // Фундаментальные и прикладные исследования в современном мире. 2014. Т. 1. № 7. С. 126-128.

4. Ямалетдинова А.М., Медведева А.С. Современные информационные и коммуникационные технологии в учебном процессе // Вестник Башкирского университета. 2016. Т. 21. № 4. С. 1134-1142.