

Нормальная форма

Нормальная форма — требование, предъявляемое к структуре таблиц в теории реляционных баз данных для устранения из базы избыточных функциональных зависимостей между атрибутами (полями таблиц).

Потенциальный ключ

- **Потенциальный ключ** - это подмножество атрибутов отношения, удовлетворяющее требованиям уникальности и минимальности (несократимости).
- **Уникальность** означает, что нет и не может быть двух кортежей данного отношения, в которых значения этого подмножества атрибутов совпадают (равны). Свойство уникальности определяется не для конкретного значения переменной отношения в тот или иной момент времени, а по всем возможным значениям, то есть следует из внешнего знания о природе и закономерностях данных, которые могут находиться в переменной отношения.
- **Минимальность (несократимость)** означает, что в составе потенциального ключа отсутствует меньшее подмножество атрибутов, удовлетворяющее условию уникальности. Иными словами, если из потенциального ключа убрать любой атрибут, он утратит свойство уникальности.

Транзитивная функциональная зависимость

Неключевой атрибут отношения R — это атрибут, который не принадлежит ни одному из потенциальных ключей R .

Функциональная зависимость множества атрибутов Z от множества атрибутов X ($X \rightarrow Z$) является транзитивной, если существует такое множество атрибутов Y , что $X \rightarrow Y$ и $Y \rightarrow Z$. При этом ни одно из множеств X , Y и Z не является подмножеством другого, то есть функциональные зависимости $X \rightarrow Z$, $X \rightarrow Y$ и $Y \rightarrow Z$ не являются тривиальными.

Функциональная зависимость *тривиальна* тогда и только тогда, когда ее правая (**зависимая**) часть является подмножеством ее левой части (**детерминанта**).

Функциональная зависимость называется *неприводимой слева*, если ни один атрибут не может быть опущен из её детерминанта без нарушения зависимости (детерминант избыточен).

Нормальные формы

Первая нормальная форма (1NF)

Вторая нормальная форма (2NF)

Третья нормальная форма (3NF)

Нормальная форма Бойса — Кодда (BCNF)

Четвёртая нормальная форма (4NF)

Пятая нормальная форма (5NF)

Доменно-ключевая нормальная форма (DKNF)

Шестая нормальная форма (6NF)

Первая нормальная форма (1NF)

Переменная отношения находится в первой нормальной форме тогда и только тогда, когда в любом допустимом значении отношения каждый его кортеж содержит только одно значение для каждого из атрибутов

Первая нормальная форма (1NF)

- Нет упорядочивания строк (порядок строк не несет в себе никакой информации).
- Нет упорядочивания столбцов
- Нет повторяющихся строк.
- Каждое пересечение строки и столбца содержит ровно одно значение из соответствующего домена (и больше ничего).
- Все столбцы являются обычными

Методы приведения к 1NF

- Устраните повторяющиеся группы в отдельных таблицах (одинаковые строки).
- Создайте отдельную таблицу для каждого набора связанных данных.
- Идентифицируйте каждый набор связанных данных с помощью первичного ключа (добавить уникальный id для каждой строки)

Вторая нормальная форма (2NF)

Переменная отношения находится во второй нормальной форме тогда и только тогда, когда она находится в первой нормальной форме и каждый неключевой атрибут неприводимо зависит от её потенциального ключа.

Неприводимость означает, что в составе потенциального ключа отсутствует меньшее подмножество атрибутов, от которого можно также вывести данную функциональную зависимость. Для неприводимой функциональной зависимости часто используется эквивалентное понятие «полная функциональная зависимость».

Методы приведения к 2NF

- Создайте отдельные таблицы для наборов значений, относящихся к нескольким записям
- Свяжите эти таблицы с помощью внешнего ключа

Третья нормальная форма (3NF)

Переменная отношения R находится в 3NF тогда и только тогда, когда выполняются следующие условия:

- R находится во второй нормальной форме.
- ни один неключевой атрибут R не находится в транзитивной функциональной зависимости от потенциального ключа R .

Нормальная форма Бойса — Кодда (BCNF)

Переменная отношения находится в BCNF тогда и только тогда, когда каждая её нетривиальная и неприводимая слева функциональная зависимость имеет в качестве своего детерминанта некоторый потенциальный ключ.

Переменная отношения находится в нормальной форме Бойса-Кодда тогда и только тогда, когда детерминанты всех её функциональных зависимостей являются потенциальными ключами.

Вынести в отдельную таблицу потенциальные первичные ключи.

Четвёртая нормальная форма (4NF)

Переменная отношения R находится в четвёртой нормальной форме, если она находится в BCNF и все нетривиальные многозначные зависимости фактически являются функциональными зависимостями от её потенциальных ключей.

Пятая нормальная форма (5NF)

Отношение находится в пятой нормальной форме (иначе — в проекционно-соединительной нормальной форме) тогда и только тогда, когда каждая нетривиальная зависимость соединения в нём определяется потенциальным ключом (ключами) этого отношения.

Нужно провести все возможные декомпозиции без потерь.

Зависимость соединения

Зависимость соединения является предельным обобщением понятий многозначной и функциональной зависимости, то есть это наиболее общая форма зависимости между атрибутами отношения

Зависимость соединения определяется не механически по текущим значениям, а следует из внешнего знания о природе и закономерностях данных, которые могут находиться в переменной отношения.

Доменно-ключевая нормальная форма (DKNF)

Переменная отношения находится в ДКНФ тогда и только тогда, когда каждое наложенное на неё ограничение является логическим следствием ограничений доменов и ограничений ключей, наложенных на данную переменную отношения.

Шестая нормальная форма (6NF)

- Переменная отношения находится в шестой нормальной форме тогда и только тогда, когда она удовлетворяет всем нетривиальным зависимостям соединения. Из определения следует, что переменная находится в 6НФ тогда и только тогда, когда она неприводима, то есть не может быть подвергнута дальнейшей декомпозиции без потерь. Каждая переменная отношения, которая находится в 6НФ, также находится и в 5НФ.

JOIN

```
SELECT field_name [,... n]  
FROM Table1 {INNER | {LEFT | RIGHT | FULL}  
    OUTER | CROSS }  
JOIN Table2  
    {ON <condition> |  
USING (field_name [,... n])}
```

INNER JOIN

Оператор внутреннего соединения **INNER JOIN** соединяет две таблицы. Порядок таблиц для оператора неважен, поскольку оператор является симметричным.

Заголовок таблицы-результата является объединением (конкатенацией) заголовков соединяемых таблиц.

INNER JOIN

```
SELECT * FROM Person  
INNER JOIN City  
ON Person.CityId = City.Id
```

OUTER JOIN

Соединение двух таблиц, в результате которого в обязательном порядке входят строки либо одной, либо обеих таблиц.

LEFT JOIN

Оператор левого внешнего соединения LEFT OUTER JOIN соединяет две таблицы. Порядок таблиц для оператора важен, поскольку оператор не является симметричным.

Заголовок таблицы-результата является объединением (конкатенацией) заголовков соединяемых таблиц.

LEFT JOIN

```
SELECT * FROM Person  
LEFT OUTER JOIN City  
ON Person.CityId = City.Id
```

Person.Name	Person.CityId	City.Id	City.Name
Андрей	1	1	Москва
Леонид	2	2	Санкт-Петербург
Сергей	1	1	Москва
Григорий	4	NULL	NULL

RIGHT JOIN

```
SELECT * FROM Person  
RIGHT OUTER JOIN City  
ON Person.CityId = City.Id
```

Person.Name	Person.CityId	City.Id	City.Name
Андрей	1	1	Москва
Сергей	1	1	Москва
Леонид	2	2	Санкт-Петербург
NULL	NULL	3	Казань

FULL OUTER JOIN

```
SELECT * FROM Person  
FULL OUTER JOIN City  
ON Person.CityId = City.Id
```

Person.Name	Person.CityId	City.Id	City.Name
Андрей	1	1	Москва
Сергей	1	1	Москва
Леонид	2	2	Санкт-Петербург
NULL	NULL	3	Казань
Григорий	4	NULL	NULL

CROSS JOIN

```
SELECT * FROM Person  
CROSS JOIN City
```

```
SELECT *  
FROM Person, City
```