Алгоритм нормализации

Уровни моделирования

При разработке БД выделяют несколько уровней моделирования:

- Сама предметная область;
- Модель предметной области;
- Логическая модель данных;
- Физическая модель данных;
- Собственно база данных и приложения.

Качество будущей БД

Основные решения закладываются на этапе разработки логической МД.

«Хорошие» МД должны удовлетворять критериям:

- адекватности БД предметной области;
- легкости разработки и сопровождения БД;
- скорости выполнения операций обновления данных (вставка, обновление, удаление);
- скорости выполнения операций выборки данных.

При создании логической модели **обычно** требуется реализовать 3 шага алгоритма нормализации

1. *Приведение к 1НФ*;
2. *Приведение к 2НФ*;
3. *Приведение к 3НФ*.

Шаг 1 (Приведение к 1НФ)

- Задается одно или несколько отношений, отображающих понятия предметной области.
- По модели предметной области (**не по внешнему виду полученных отношений!!!**) выписываются обнаруженные ФЗ.
- **Все отношения автоматически находятся в 1НФ**.

1НФ - это обычное отношение. Отношение в 1НФ обладает следующими свойствами:

- в отношении нет одинаковых кортежей;
- кортежи не упорядочены;
- атрибуты не упорядочены;
- все значения атрибутов атомарны.

Шаг 2 (Приведение к 2НФ)

Если в отношении обнаружена зависимость атрибутов от части сложного (составного) ключа, то проводим его **декомпозицию**:

- те атрибуты, которые зависят от части сложного ключа выносятся в отдельное отношение вместе с этой частью ключа;
- в исходном отношении остаются все ключевые атрибуты.

Шаг 2 (Приведение к 2НФ)

Исходное отношение: $R(K_1, K_2, A_1, .., A_n, B_1, \ldots, B_m)$

$\{K_1, K_2\}$ - сложный (составной) ключ.

Функциональные зависимости:

$\{K_1, K_2\} \rightarrow \{A_1, .., A_n, B_1, \ldots B_m\}$ - зависимость всех атрибутов от ключа отношения.

$\{K_1\} \rightarrow \{A_1, .., A_n\}$ - зависимость некоторых атрибутов от части сложного ключа.

Шаг 2 (Приведение к 2НФ)

Декомпозированные отношения:

$R_1(K_1, K_2, B_1, \ldots, B_m)$ - остаток от исходного отношения

Ключ - (K_1, K_2)

$R_2(K_1, A_1, .., A_n)$ - атрибуты, вынесенные из исходного отношения вместе с частью сложного ключа.

Ключ - K_1

Вывод

Отношение находится в **2НФ** тогда и только тогда, когда отношение находится в 1НФ и нет неключевых атрибутов, зависящих от части сложного ключа.

Шаг 3 (Приведение к 3НФ)

Если в отношении обнаружена зависимость одних неключевых атрибутов от других неключевых атрибутов, то проводим декомпозицию:

- те неключевые атрибуты, которые зависят от других неключевых атрибутов, выносятся в отдельное отношение;
- в новом отношении ключом становится детерминант функциональной зависимости.

Шаг 3 (Приведение к 3НФ)

Исходное отношение: $R(K, A_1, .., A_n, B_1, \ldots, B_m)$

Ключ - K

Функциональные зависимости:

$K \rightarrow \{A_1, .., A_n, B_1, \ldots, B_m\}$ - зависимость всех атрибутов от ключа отношения.

$\{A_1, .., A_n\} \rightarrow \{B_1, \ldots, B_m\}$ - зависимость одних неключевых атрибутов от других неключевых атрибутов.

Шаг 3 (Приведение к 3НФ)

Декомпозированные отношения:

$R_1(K, A_1, .., A_n)$ - остаток от исходного отношения.

Ключ - K

$R_2(A_1, .., A_n, B_1, \ldots, B_m)$ - атрибуты, вынесенные из исходного отношения вместе с детерминантом функциональной зависимости.

Ключ - $\{A_1, .., A_n\}$

Вывод

Отношение находится в **3НФ** тогда и только тогда, когда отношение находится в 2НФ и все неключевые атрибуты взаимно независимы.

Отношения в 3НФ являются самыми «хорошими» с точки зрения выбранных критериев - устранены аномалии обновления, требуются только стандартные триггеры для поддержания ссылочной целостности.

Замечание

На практике опытные разработчики, как правило, сразу строят отношения в 3НФ.

Для разработки логических моделей данных применяют различные варианты ER-диаграмм.

Особенность этих диаграмм в том, что они сразу позволяют создавать отношения в 3НФ.

Приведенный алгоритм важен по двум причинам

1) этот алгоритм показывает, какие проблемы возникают при разработке слабо нормализованных отношений;

2) как правило, модель предметной области никогда не бывает правильно разработана с первого шага.

Пояснения: 1) Эксперты предметной области могут забыть о чем-либо упомянуть, 2) разработчик может неправильно понять эксперта, 3) во время разработки могут измениться правила, принятые в предметной области, и т.д.

$\Rightarrow$ могут появиться новые зависимости, которые отсутствовали в первоначальной модели предметной области $\Rightarrow$ необходимо использовать алгоритм нормализации, чтобы убедиться, что отношения остались в 3НФ и логическая модель не ухудшилась.

Рассмотренный алгоритм нормализации заключается в последовательной декомпозиции отношений для устранения ФЗ атрибутов от части сложного ключа (приведение к 2НФ) и устранения ФЗ неключевых атрибутов друг от друга (приведение к 3НФ).

Продолжение алгоритма нормализации

4. *Приведение к НФБК;*
5. *Приведение к 4НФ;*
6. *Приведение к 5НФ.*

Шаг 4 (Приведение к НФБК)

4.1. Если отношения содержат несколько потенциальных ключей, то необходимо проверить, имеются ли ФЗ, детерминанты которых не являются потенциальными ключами.

4.2. Если такие ФЗ имеются, то необходимо провести дальнейшую декомпозицию отношений.

4.3. Те атрибуты, которые зависят от детерминантов, не являющихся потенциальными ключами, выносятся в отдельное отношение вместе с детерминантами.

Шаг 5 (Приведение к 4НФ)

Если в отношениях обнаружены **нетривиальные многозначные зависимости**, то необходимо провести декомпозицию для исключения таких зависимостей.

Шаг 6 (Приведение к 5НФ)

Если в отношениях обнаружены **нетривиальные зависимости соединения**, то необходимо провести декомпозицию для исключения таких зависимостей.

Подведем итоги

1. Алгоритм нормализации состоит в выявлении ФЗ **предметной области** и соответствующей декомпозиции отношений.
2. Для решения большинства практических задач **достаточно** получения 3НФ.
3. Обобщением 3НФ на случай, когда отношение имеет более одного потенциального ключа, является **НФБК**.
4. Отношение находится в НФБК тогда и только тогда, когда **детерминанты** всех ФЗ являются потенциальными ключами.

Подведем итоги

5. Нормализация отношений вплоть до НФБК основана на понятии ФЗ и **теореме Хеза**, гарантировавшей, что декомпозиция будет происходить без потерь информации.
6. Дальнейшая нормализация связана уже с **обобщением** понятия ФЗ.
7. Корректность дальнейшей декомпозиции основывается на **теореме Фейджина** - декомпозиция отношения на две проекции является декомпозицией без потерь тогда и только тогда, когда в отношении имеется некоторая МЗЗ.

Подведем итоги

8. Если в отношении имеется ФЗ, то автоматически имеется и **тривиальная** МЗЗ, определяемая этой ФЗ.

9. Имеются зависимости специального вида, когда отношение не может быть подвергнуто декомпозиции без потерь на две проекции, но может быть декомпозировано на большее число проекций. Такие зависимости называются **зависимостями соединения** и являются обобщением понятия МЗЗ.

Сравнение нормализованных и ненормализованных моделей

Критерий	Отношения слабо нормализованы (1НФ, 2НФ)	Отношения сильно нормализованы (3НФ)
Адекватность БД предметной области	ХУЖЕ (-)	ЛУЧШЕ (+)
Легкость разработки и сопровождения БД	СЛОЖНЕЕ (-)	ЛЕГЧЕ (+)
Скорость выполнения вставки, обновления, удаления	МЕДЛЕННЕЕ (-)	БЫСТРЕЕ (+)
Скорость выполнения выборки данных	БЫСТРЕЕ (+)	МЕДЛЕННЕЕ (-)

Проанализируем влияние логического моделирования данных на качество физических моделей данных и производительность БД.

Результат

- **Сильно нормализованные отношения** оказываются лучше спроектированы (3+, 1-). Они больше соответствуют предметной области, легче в разработке, для них быстрее выполняются операции модификации БД. Но это достигается ценой некоторого замедления выполнения операций выборки данных.
- **У слабо нормализованных отношений** единственное преимущество - если к БД обращаться только с запросами на выборку данных, то такие запросы выполняются быстрее. Это связано с тем, что в таких отношениях не тратится время на соединение отношений.
- **Выбор степени нормализации отношений** зависит от характера запросов, с которыми чаще всего обращаются к базе данных.

Спасибо за внимание!