

Лабораторная работа № 6. Вариативная часть

Вариант 1. Проанализировать информацию по избыточности баз данных и привести примеры неверного логического проектирования. Предложить алгоритм решения (по устранению) недостатков логической структуры.

Решение:

Избыточность полезна, например, для резервного копирования. Но она также нежелательна:

- Увеличивает объем хранимых данных.
- Затрудняет обновление данных.
- Снижает производительность: запросы к базе данных могут занимать больше времени, так как требуется обработка большего объема данных.
- Увеличивает риск аномалий при вставке, удалении и обновлении данных.

Примеры неверного логического проектирования:

Предположим, у нас есть небольшая библиотека. Изначально мы хотим хранить информацию о книгах и авторах в одной таблице.

Исходная, ненормализованная таблица «КнигиИАвторы»

ID_Книг и	НазваниеКниг и	АвторИм я	АвторФамили я	АвторСтран а	ЖанрКниг и
1	“Мастер и Маргарита”	Михаил	Булгаков	СССР	Роман
2	“Собачье сердце”	Михаил	Булгаков	СССР	Сатира
3	“1984”	Джордж	Оруэлл	Англия	Антиутопия
4	“Скотный двор”	Джордж	Оруэлл	Англия	Сатира
5	“Война и мир”	Лев	Толстой	Россия	Роман

Проблемы:

- **Избыточность информации об авторе:** Имя, фамилия и страна автора повторяются для каждой книги этого автора. В примере с Булгаковым и Оруэллом это видно очень четко.
- **Проблемы с обновлением:** Если автор сменит страну (например, переедет), придется обновлять все записи книг этого автора.
- **Риск несогласованности:** Возможно случайное внесение разных вариантов имени/фамилии одного и того же автора (например, “Михаил Булгаков” и “М. Булгаков”).
- **Увеличение объема хранения:** Дублирование информации занимает лишнее место.
- **Аномалия вставки:** Невозможно добавить информацию об авторе, пока у него не будет написана книга.
- **Аномалия удаления:** Удаление последней книги автора приведет к потере информации об авторе.

Алгоритм решения (устранения) недостатков логической структуры (нормализация):

Алгоритм решения проблем избыточности в базах данных обычно сводится к процессу, называемому нормализацией. Нормализация — это процесс организации данных в базе данных для уменьшения избыточности и повышения целостности данных. Он включает в себя разделение больших таблиц на меньшие и определение связей между ними. Существуют различные нормальные формы, каждая из которых устраняет определенный вид избыточности.

1. **Определите атрибуты (поля):** составьте список всех атрибутов, которые необходимо хранить в базе данных.
2. **Определите функциональные зависимости:** установите, какие атрибуты функционально зависят от других атрибутов. Атрибут В функционально зависит от атрибута А, если каждое значение А однозначно определяет значение В.
3. **Первая Нормальная Форма (1NF):**
 - Устраните повторяющиеся группы атрибутов.
 - Создайте отдельную таблицу для каждой группы связанных атрибутов.
 - Определите первичный ключ для каждой таблицы.

Пример: Таблица «КнигиИАвторы» уже находится в 1NF, так как в ней нет повторяющихся групп. Каждое поле содержит атомарное значение.

4. Вторая Нормальная Форма (2NF):

- Таблица должна быть в 1NF.
- Устраните неполную зависимость от первичного ключа. Атрибуты, зависящие только от части составного первичного ключа, должны быть вынесены в отдельную таблицу.

Пример: Разделяем информацию о книгах и авторах в отдельные таблицы.

- Авторы: (ID_Авторы, АвторИмя, АвторФамилия, АвторСтрана)
- Книги: (ID_Книги; НазваниеКниги; ID_Автора; ЖанрКниги)

5. Третья Нормальная Форма (3NF):

- Таблица должна быть в 2NF.
- Устраните транзитивную зависимость. Атрибуты, зависящие от неключевого атрибута, должны быть вынесены в отдельную таблицу.

Пример: если у нас, к примеру, есть дополнительное поле в таблице Книги - ОписаниеЖанра, которое зависит только от ЖанрКниги, тогда можно выделить Жанры в отдельную таблицу:

- Жанры: (ID_Жанра, ЖанрКниги, ОписаниеЖанра)
- Авторы: (ID_Автора, АвторИмя, АвторФамилия, АвторСтрана)
- Книги: (ID_Книги, НазваниеКниги, ID_Автора, ID_Жанра)

6. Четвертая нормальная форма (4NF) и пятая нормальная форма (5NF): эти нормальные формы рассматривают более сложные случаи зависимостей, такие как многозначные и составные зависимости. Они используются реже, чем 1NF, 2NF и 3NF.