

ЛАБОРАТОРНАЯ РАБОТА № 2

Темы: Этапы проектирования базы данных. Дублирование и аномалии. Нормализация отношений. Функциональные зависимости. Нормальные формы.

Проектирование базы данных

Проектирование базы данных представляет собой процесс создания структуры базы данных, позволяющей эффективно хранить и обрабатывать информацию. Процесс включает несколько этапов:

Этап 1: Анализ требований

Цель этапа заключается в сборе и анализе потребностей пользователей относительно хранения и обработки данных. Формируются требования к информационной системе, определяются объекты предметной области и связи между ними.

Этап 2: Концептуальное проектирование

На данном этапе создается концептуальная модель данных (ER-модель). Определяются сущности, атрибуты и отношения между сущностями. Строится диаграмма ER-модели.

Этап 3: Логическое проектирование

Концептуальная модель преобразуется в логическую структуру базы данных. Для реляционных СУБД строится схема отношений. Выявляются первичные ключи, внешние ключи и ограничения целостности.

Этап 4: Физическое проектирование

Логическая структура базы данных адаптируется к конкретной СУБД. Создаются таблицы, индексы, триггеры и хранимые процедуры.

Этап 5: Реализация и тестирование

Создается физическая база данных, загружаются тестовые данные, проводится проверка работоспособности системы.

Дублирование и аномалии

При проектировании базы данных важно избегать избыточности и аномалий обновления. Рассмотрим пример:

Студент	Предмет	Оценка
Иванов	Математика	5

Петров	История	4
Сидоров	Физика	3

В приведенной таблице наблюдается проблема дублирования данных: каждому студенту соответствует отдельная строка для каждого предмета. Например, если мы решим добавить результаты Иванова по истории, нам придется создать новую строку, в которой фамилию Иванов придется вводить заново. Это ведет к потенциальным проблемам:

Аномалия вставки: нельзя добавить нового студента, пока не известны предметы и оценки.

Аномалия удаления: удаление строки удалит всю информацию о студенте вместе с оценками.

Аномалия изменения: изменение имени студента потребует обновить каждую строку отдельно.

Для устранения проблемы следует нормализовать таблицу, разделив её на две части:

Таблица студентов

ID_Студента	Имя
1	Иванов
2	Петров
3	Сидоров

Таблица оценок

ID_Студента	Предмет	Оценка
1	Математика	5
2	История	4
3	Физика	3

Теперь каждый студент представлен отдельной записью, а связь между студентами и предметами осуществляется через внешний ключ (ID_Студента).

Нормализация отношений

Процесс нормализации позволяет устранить избыточность и повысить эффективность базы данных путем разделения таблиц на меньшие, менее взаимосвязанные компоненты. Выделяют несколько нормальных форм (здесь приведены только 5 из них):

Первая нормальная форма (1NF): Таблицы содержат атомарные значения, отсутствуют повторяющиеся группы столбцов.

Вторая нормальная форма (2NF): Все неключевые атрибуты зависят от полного ключа, устраняются частичные функциональные зависимости.

Третья нормальная форма (3NF): Устранены транзитивные зависимости, все неключевые атрибуты зависят непосредственно от ключа.

Четвертая нормальная форма (4NF): Устраняются зависимости множественного выбора, приводящие к дубликам данных.

Пятая нормальная форма (5NF): Разделены таблицы таким образом, чтобы исключить любые возможные потери информации при объединениях.

Функциональные зависимости

Функциональная зависимость возникает тогда, когда значение одного атрибута однозначно определяет значение другого атрибута. Например, если известно ID_Студента, мы можем определить его имя. Записывается функциональная зависимость следующим образом:

ID_Студента \rightarrow Имя

Знание функциональной зависимости помогает выявить проблемные места в структуре базы данных и провести нормализацию.

Иными словами, множество В функционально зависит от А (обозначается $A \rightarrow B$) тогда и только тогда, когда каждое значение множества А связано в точности с одним значением множества В.

Задание для выполнения

1. Преобразуйте концептуальную модель вашей базы данных (ER-модель) в логическую, постройте схему отношений. Выявите первичные ключи, внешние ключи и ограничения целостности.
2. Определите есть ли дублирование и аномалии в вашей базе данных. Какие именно из них присутствуют?
3. Определите каким нормальным формам соответствуют таблицы в вашей базе данных.
4. Определите функциональные зависимости в вашей базе данных.
5. Там, где это требуется, нормализуйте таблицы. Ограничьтесь 3 или 4 нормальной формой. Сравните полученный результат и таблицы до нормализации.
6. Как изменилась логическая модель данных после преобразований (нормализации)?
7. Проверить полноту и корректность логической модели базы данных путём составления на языке SQL типовых запросов для поиска и анализа информации (10-20 запросов).
8. Отметьте, в чем заключаются положительные и отрицательные стороны процедуры нормализации применительно к вашей базе данных.

Общая информация.

1. Архитектура базы данных

Архитектура базы данных определяет структуру и организацию компонентов системы, обеспечивающих эффективное управление данными. Обычно выделяют три уровня архитектуры базы данных:

- Внешний уровень: представление данных для конечных пользователей, определяется интерфейсами и приложениями.
- Концептуальный уровень: общее описание данных и связей между ними, отражающее бизнес-требования и логику предметной области.
- Физический уровень: физическая реализация базы данных, включающая физическую структуру файлов, индексы, оптимизацию производительности.

2. Нормализация данных

Нормализация — процесс структурирования данных таким образом, чтобы минимизировать избыточность и повысить целостность данных. Существует несколько нормальных форм нормализации (приведены только 4 из них):

- Первая нормальная форма (1NF): каждое поле таблицы должно содержать атомарные (неделимые) значения.
- Вторая нормальная форма (2NF): таблица должна находиться в первой нормальной форме, и все неключевые атрибуты зависят от полного ключа.
- Третья нормальная форма (3NF): таблица находится во второй нормальной форме, и ни одно значение не зависит от другого неключевого атрибута.
- Четвертая нормальная форма (BCNF, Boyce-Codd Normal Form): дополнительное требование, устраняющее зависимости между атрибутами, возникающими вследствие неполных функциональных зависимостей.

3. Транзакции и ACID-принципы

Транзакция — последовательность операций над базой данных, выполняемых как единое целое. Принцип ACID гарантирует надёжность транзакций:

- Atomicity (атомарность): либо все операции завершаются успешно, либо не выполняется ничего.
- Consistency (согласованность): состояние базы данных остаётся согласованным перед началом и после окончания транзакции.
- Isolation (изоляция): параллельные транзакции выполняются независимо друг от друга.
- Durability (устойчивость): выполненные изменения сохраняются даже в случае сбоя оборудования или ПО.

4. Индексация и производительность

Индексы ускоряют поиск данных в больших объёмах. Они представляют собой дополнительные структуры данных, хранящие упорядоченные значения определённых столбцов вместе с указанием местоположения соответствующих записей.

Виды индексов:

- Простой индекс: создаётся на одном столбце.
- Составной индекс: включает два или более столбца.
- Первичный индекс: строится на первичном ключе.

- Обратный индекс: ускоряет полнотекстовый поиск.

Правильно настроенные индексы значительно повышают производительность выборки, однако требуют дополнительного пространства и увеличивают затраты на обновления.

5. Безопасность баз данных

Безопасность баз данных включает защиту данных от несанкционированного доступа, модификаций и утечек. Основные меры защиты:

- Аутентификация и авторизация: проверка подлинности пользователей и контроль прав доступа.
- Шифрование: защита данных при хранении и передаче.
- Резервное копирование: регулярное сохранение копий данных для восстановления после сбоев.
- Мониторинг активности: отслеживание действий пользователей и выявление подозрительных действий.

НОРМАЛИЗАЦИЯ

Процесс нормализации является ключевым этапом проектирования реляционных баз данных. Его цель — привести структуру базы данных к виду, минимизирующему дублирование данных и обеспечивающему целостность данных. Основная задача состоит в том, чтобы избежать избыточности и обеспечить максимальную эффективность обработки данных.

Понятие нормализации и нормальных форм

Нормализация представляет собой последовательный процесс приведения базы данных к определенным стандартам, называемым нормальными формами (НФ). Каждая последующая НФ устраняет недостатки предыдущей, обеспечивая лучшее структурирование данных и снижение риска возникновения ошибок.

Существует несколько стандартных уровней нормальных форм, каждый из которых накладывает дополнительные ограничения на структуру базы данных:

1. Первая нормальная форма (1NF)

Первая нормальная форма требует, чтобы каждое поле каждой записи содержало одно значение и было атомарным (неделимым). То есть в таблице не должно быть составных полей, содержащих множественные значения (например, список телефонов, записанный одной строкой).

2. Вторая нормальная форма (2NF)

Вторая нормальная форма достигается тогда, когда база данных находится в первой нормальной форме, и все поля зависят от всего первичного ключа, а не от его части. Эта форма предотвращает возникновение проблем с избыточностью данных, вызванных наличием частично функциональных зависимостей.

3. Третья нормальная форма (3NF)

Третья нормальная форма предполагает устранение транзитивных зависимостей. Транзитивная зависимость возникает, когда два атрибута связаны друг с другом опосредованно через третий

атрибут. Чтобы перейти к 3NF, необходимо убедиться, что ни один атрибут не зависит от другого через промежуточный атрибут.

4. Четвертая нормальная форма (4NF)

Четвёртая нормальная форма касается устранения многосторонних зависимостей. Она применяется, когда одна сущность имеет связи сразу с несколькими другими типами объектов, вызывая потенциальные конфликты целостности данных.

5. Пятая нормальная форма (5NF)

Пятую нормальную форму называют также проекционно-соединительной формой. Ее основное назначение — предотвращение потери данных при соединениях таблиц. Достигается путём исключения сложных связей между элементами, приводящих к утрате значений при выполнении операций соединения.

Кроме перечисленных классических нормальных форм существуют и расширенные варианты, например, доменно-ключевая нормальная форма (DK/NF), однако их использование редко встречается на практике.

Процесс нормализации баз данных

Основные шаги процесса нормализации включают:

1. Создание первоначального проекта базы данных. Определяются сущности и отношения между ними, составляется предварительная структура таблиц.
2. Приведение к первой нормальной форме (1NF). Удаляются повторяющиеся элементы и составные поля, вводятся однозначные первичные ключи.
3. Переход ко второй нормальной форме (2NF). Устраняются частичные функциональные зависимости, создаются отдельные таблицы для отдельных сущностей.
4. Переработка до третьей нормальной формы (3NF). Избавляются от транзитивных зависимостей, обеспечивается независимость несвязанных элементов.
5. Проверка на соответствие четвертой и пятой нормальным формам, если требуется достижение высокого уровня нормализации.

Преимущества нормализации

Правильная нормализация базы данных обеспечивает следующие преимущества:

- Минимизация избыточности данных.
- Улучшение производительности запросов благодаря отсутствию ненужных данных.
- Повышенная надежность и простота поддержки структуры данных.
- Предотвращение аномалий вставки, удаления и модификации данных.

Недостатки нормализации

Несмотря на очевидные достоинства, чрезмерная степень нормализации может приводить к некоторым проблемам:

- Увеличению числа соединений (JOINS) при выполнении запросов, что снижает производительность системы.
- Сложности понимания структуры данных новичком.
- Дополнительному объему работ по поддержке большого числа мелких таблиц.

Поэтому на практике часто применяют компромиссные решения, балансируя между уровнем нормализации и удобством эксплуатации базы данных.