

Реляционные СУБД.

Реляционная СУБД (РСУБД; иначе Система управления реляционными базами данных, СУРБД) — СУБД, управляющая реляционными базами данных.

Понятие реляционный ([англ. relation](#) — отношение) связано с разработками известного английского специалиста в области систем баз данных [Эдгара Кодда](#) (Edgar Codd).

Эти модели характеризуются простотой структуры данных, удобным для пользователя табличным представлением и возможностью использования формального аппарата алгебры отношений и реляционного исчисления для обработки данных.

Реляционная модель ориентирована на организацию данных в виде двумерных таблиц. Каждая реляционная таблица представляет собой двумерный массив и обладает следующими свойствами:

- КАЖДЫЙ ЭЛЕМЕНТ ТАБЛИЦЫ — ОДИН ЭЛЕМЕНТ ДАННЫХ
- ВСЕ СТОЛБЦЫ В ТАБЛИЦЕ ОДНОРОДНЫЕ, ТО ЕСТЬ ВСЕ ЭЛЕМЕНТЫ В СТОЛБЦЕ ИМЕЮТ ОДИНАКОВЫЙ ТИП (ЧИСЛОВОЙ, СИМВОЛЬНЫЙ И Т. Д.)
- КАЖДЫЙ СТОЛБЕЦ ИМЕЕТ УНИКАЛЬНОЕ ИМЯ
- ОДИНАКОВЫЕ СТРОКИ В ТАБЛИЦЕ ОТСУТСТВУЮТ
- ПОРЯДОК СЛЕДОВАНИЯ СТРОК И СТОЛБЦОВ МОЖЕТ БЫТЬ ПРОИЗВОЛЬНЫМ

Базовыми понятиями реляционных СУБД являются: 1) АТТРИБУТ 2) [ОТНОШЕНИЯ](#) 3) [КОРТЕЖ](#)

ИНДЕКСЫ. ПРИМЕНЕНИЕ ИНДЕКСОВ.

Что такое индекс? Специальный аппарат ускоряющий обработку данных. В настоящее время в базах по умолчанию доступны B-TREE/HASH индексы. Если почитать документацию к базе можно выяснить как именно применяются индексы, вот несколько общих мест применения:

SELECT * FROM T AS t1 WHERE t1.i1 > CONST; i1 — индексированное поле таблицы T. В данном случае вместо сканирования полной таблицы база сразу отсекает ненужные значения.

Использование индексов можно проверить командами EXPLAIN/DESCRIBE ETC. В случае если предложенные базой индексы не устраивают разработчика в SQL запросах как правило предусмотрены места для подсказок базе данных.

В настоящее время MySQL например позволяет использовать R-INDEX для GEO данных. Данный индекс есть оптимизатор поиска географических объектов.

ORACLE добавляет частичные индексы для применения различных стратегий индексирования разным значениям поля.

Транзакции. Свойства транзакций.

Транзакция ([англ. TRANSACTION](#)) — в [информатике](#), группа последовательных операций, которая представляет собой логическую единицу работы с данными. Транзакция может быть выполнена целиком либо успешно, соблюдая целостность данных и независимо от параллельно идущих других транзакций, либо не выполнена вообще и тогда она не должна произвести никакого эффекта. Транзакции обрабатываются [транзакционными системами](#), в процессе работы которых создаётся [история транзакций](#). Обычные и [распределённые транзакции](#). Распределённые транзакции подразумевают использование больше чем одной транзакционной системы и требуют намного более сложной логики (например, TWO-PHASE COMMIT — [двухфазный протокол подтверждения успеха](#)). Также, в некоторых системах реализованы [автономные транзакции](#), или под-транзакции, которые являются автономной частью родительской транзакции.

Известны также как свойства [ACID](#) (ATOMICITY — неделимость, CONSISTENCY —

СОГЛАСОВАННОСТЬ, ISOLATION — изоляция, DURABILITY — устойчивость) транзакций.

1. Транзакция неделима в том смысле, что представляет собой единое целое. Все её компоненты либо имеют место, либо нет. Не бывает частичной транзакции. Если может быть выполнена лишь часть транзакции, она отклоняется.
2. Транзакция является согласованной, потому что не нарушает бизнес-логику и отношения между элементами данных. Это свойство очень важно при разработке клиент-серверных систем, поскольку в хранилище данных поступает большое количество транзакций от разных систем и объектов. Если хотя бы одна из них нарушит целостность данных, то все остальные могут выдать неверные результаты.
3. Транзакция всегда изолирована, поскольку её результаты самодостаточны. Они не зависят от предыдущих или последующих транзакций — это свойство называется сериализуемостью и означает, что транзакции в последовательности независимы.
4. Транзакция устойчива. После своего завершения она сохраняется в системе, которую ничто не может вернуть в исходное (до начала транзакции) состояние, то есть происходит фиксация транзакции, означающая, что её действие постоянно даже при сбое системы. При этом подразумевается некая форма хранения информации в постоянной памяти как часть транзакции.

Уровни изоляции транзакций

В идеале транзакции разных пользователей должны выполняться так, чтобы создавалась иллюзия, что пользователь текущей транзакции — единственный. Однако, в реальности, по соображениям производительности и для выполнения некоторых специальных задач, СУБД предоставляют различные уровни изоляции транзакций. Уровни описаны в порядке увеличения изоляции транзакций и надёжности работы с данными

- **0 — Неподтверждённое чтение** (READ UNCOMMITTED, DIRTY READ, грязное чтение) — чтение незафиксированных изменений своей транзакции и конкурирующих транзакций, возможны нечистые, неповторяемые чтения и фантомы
- **1 — Подтверждённое чтение** (Read Committed) — чтение всех изменений своей транзакции и зафиксированных изменений конкурирующих транзакций, нечистые чтения невозможны, возможны неповторяемые чтения и фантомы
- **2 — Повторяемое чтение** (REPEATABLE READ, SNAPSHOT) — чтение всех изменений своей транзакции, любые изменения, внесённые конкурирующими транзакциями после начала своей недоступны, нечистые и неповторяемые чтения невозможны, возможны фантомы
- **3 — Упорядоченный** — (Serializable, сериализуемый) упорядоченные (сериализуемые) транзакции. Идентичен ситуации при которой транзакции выполняются строго последовательно одна после другой. То есть транзакции, результат действия которых не зависит от порядка выполнения шагов транзакции (запрещено чтение всех данных изменённых с начала транзакции, в том числе и своей транзакцией). Фантомы невозможны.

Чем выше уровень изоляции, тем больше требуется ресурсов, чтобы их поддерживать.

В СУБД уровень изоляции транзакций можно выбрать как для всех транзакций сразу, так и для одной конкретной транзакции. По умолчанию в большинстве баз данных используется уровень 1 (READ COMMITTED). Уровень 0 используется в основном для отслеживания изменений длительных транзакций или для чтения редкоизменяемых данных. Уровни 2 и 3 используются при повышенных требованиях к изолированности транзакций.

Блокировки.

Два типа блокировок есть в природе: оптимистичная и пессимистичная. В первом случае, блокировка используется только при обновлении записи, пессимистичная же запрещает доступ к записи как только происходит доступ.