

Для PostgreSQL свойственны:

Поддержка востребованного объектно-ориентированного, реляционного подхода к базам данных.

Полная поддержка надежных транзакций, т.е. атомарность, последовательность, изоляционность, прочность.

Благодаря мощным технологиям Postgre очень производительна. Параллельность достигнута не за счет блокировки операций чтения, а благодаря реализации управления многовариантным параллелизмом (MVCC), что также обеспечивает соответствие ACID.

## **ACID**

### **Atomicity — Атомарность**

Атомарность гарантирует, что никакая транзакция не будет зафиксирована в системе частично. Будут либо выполнены все её подоперации, либо не выполнено ни одной. Если транзакцию не удаётся полностью завершить, результаты всех её до сих пор произведённых действий будут отменены и система вернётся во «внешне исходное» состояние

### **Consistency — Согласованность**

Транзакция, достигающая своего нормального завершения и тем самым фиксирующая свои результаты, сохраняет согласованность базы данных. Другими словами, каждая успешная транзакция по определению фиксирует только допустимые результаты. Это условие является необходимым для поддержки четвёртого свойства.

### **Isolation — Изолированность**[\[править | править код\]](#)

Во время выполнения транзакции параллельные транзакции не должны оказывать влияния на её результат. Изолированность — требование дорогое,

поэтому в реальных БД существуют режимы, не полностью изолирующие транзакцию (уровни изолированности Repeatable Read и ниже).

### **Durability - Прочность**

Независимо от проблем на нижних уровнях (к примеру, обесточивание системы или сбой в оборудовании) изменения, сделанные успешно завершённой транзакцией, должны остаться сохранёнными после возвращения системы в работу. Другими словами, если пользователь получил подтверждение от системы, что транзакция выполнена, он может быть уверен, что сделанные им изменения не будут отменены из-за какого-либо сбоя.

**MVCC** (управление параллельным доступом посредством многоверсионности) один из механизмов СУБД для обеспечения параллельного доступа к базам данных, заключающийся в предоставлении каждому пользователю так называемого «снимка» базы, обладающего тем свойством, что вносимые пользователем изменения невидимы другим пользователям до момента фиксации транзакции. Этот способ управления позволяет добиться того, что пишущие транзакции не блокируют читающих, и читающие транзакции не блокируют пишущих.

### **Функции**

в PostgreSQL являются блоками кода, исполняемыми на сервере, а не на клиенте БД. Хотя они могут писаться на чистом SQL, реализация дополнительной логики, например, условных переходов и циклов, выходит за рамки, собственно, SQL и требует использования некоторых языковых расширений. Функции могут писаться с использованием различных языков программирования. PostgreSQL допускает использование функций, возвращающих набор записей, который далее можно использовать так же, как и результат выполнения обычного запроса. Функции могут выполняться как с правами их создателя, так и с правами текущего пользователя. Иногда

функции отождествляются с хранимыми процедурами, однако между этими понятиями есть различие.

## **Триггеры**

в PostgreSQL определяются как функции, инициируемые DML-операциями. Например, операция INSERT может запускать триггер, проверяющий добавленную запись на соответствия определённым условиям. При написании функций для триггеров могут использоваться различные языки программирования. Триггеры ассоциируются с таблицами. Множественные триггеры выполняются в алфавитном порядке.

## **Правила**

Иногда требуется выполнить различные/альтернативные действия во время запросов к базе данных. Это может потребоваться для защиты данных или их абстракции. Система правил (rule) PostgreSQL позволяет определить альтернативные действия на вставку, обновление или удаление. Правило генерирует дополнительный запрос. Как результат, выполнение правила оказывает влияние на производительность системы.

## **Индексы**

В PostgreSQL следующих типов: B-дерево, хэш, R-дерево, GiST, GIN. При необходимости можно создавать новые типы индексов.

## **Многоверсионность**

поддерживается в PostgreSQL — возможна одновременная модификация БД несколькими пользователями с помощью механизма Multiversion Concurrency Control (MVCC). Благодаря этому соблюдаются требования ACID, и практически отпадает нужда в блокировках чтения.

## **Расширение**

PostgreSQL для собственных нужд возможно практически в любом аспекте. Есть возможность добавлять собственные преобразования типов, типы данных, домены (пользовательские типы с изначально наложенными ограничениями), функции (включая агрегатные), индексы, операторы (включая переопределение уже существующих) и процедурные языки.

## **Наследование**

в PostgreSQL реализовано на уровне таблиц. Таблицы могут наследовать характеристики и наборы полей от других таблиц (родительских). При этом данные, добавленные в порождённую таблицу, автоматически будут участвовать (если это не указано отдельно) в запросах к родительской таблице.

## **Достоинства PostgreSQL**

Открытое ПО, соответствующее стандарту SQL. PostgreSQL – бесплатное ПО с открытым исходным кодом. Эта СУБД к тому же является очень мощной системой.

У данной СУБД существует большое сообщество, у которого можно без труда найти ответы на свои вопросы

Большое количество дополнений – несмотря на огромное количество встроенных функций, существует очень много дополнений, позволяющих разрабатывать данные для этой СУБД и управлять ими.

PostgreSQL это не только реляционная СУБД, но также и объектно-ориентированная с поддержкой наследования и много другого

## **Недостатки PostgreSQL**

Производительность - при простых операциях чтения PostgreSQL может значительно замедлить сервер и быть медленнее своих конкурентов, таких как MySQL

Хостинг – иногда довольно сложно найти хостинг с поддержкой этой СУБД.

### **Когда использовать PostgreSQL**

- Если требуется масштабируемость при смешанной (чтение и запись) нагрузке. Целесообразно использовать его на сайтах с часто изменяющимися данными и большим количеством посещений
- если приоритет стоит на надёжность и целостность данных, PostgreSQL – лучший выбор.
- Если в будущем предстоит перемещать всю базу на другое решение, меньше всего проблем возникнет с PostgreSQL.
- Если БД должна выполнять сложные процедуры, стоит выбрать PostgreSQL в силу её расширяемости.

### **Когда не следует использовать PostgreSQL**

- Если быстрое чтение решающий фактор, то стоит присмотреться к другим СУБД
- Если не нужна целостность данных, соответствие ACID или сложные структуры данных, то настройка PostgreSQL может быть нецелесообразна

### **Использование в веб-проектах**

В разработке простых сайтов PostgreSQL используется несколько реже, чем MySQL / MariaDB. При этом в разработке сложных сайтов и веб-приложений PostgreSQL опережает по использованию MySQL и MariaDB.