

Реляционные базы данных: Резервное копирование.



Роман
Гордиенко



Роман Гордиенко

Backend Developer, Factory5




[Роман Гордиенко](#)



План занятия

1. [Резервное копирование — что это?](#)
2. [Виды и типы резервного копирования](#)
3. [Восстановление данных](#)
4. [Резервное копирование PostgreSQL](#)
5. [Резервное копирование MySQL](#)
6. [Мониторинг состояния бэкапов](#)
7. [Итоги](#)
8. [Домашнее задание](#)




Резервное копирование — что это?

Резервное копирование (backup)

Резервное копирование — процесс создания копии данных на носителе (жёстком диске, дискете и т. д.), предназначенном для восстановления данных в оригинальном или новом месте их расположения в случае их повреждения или разрушения.

Резервное копирование необходимо для возможности быстрого и недорогого восстановления БД в случае утери рабочей копии информации по какой-либо причине.



Виды и типы резервного копирования

Виды и типы резервного копирования

- Полное резервное копирование (Full backup);
- Инкрементные бэкапы (Incremental backup);
- Дифференциальный бэкап (differential backup);
- Холодное резервирование БД;
- Горячее резервирование БД;
- Репликация.



Полное резервное копирование (Full backup)

Данный метод создает полную копию набора исходных данных.

В данном случае копируется весь каталог с файлами БД либо создается образ диска или системы, где находится база.

Преимущества и недостатки полного резервного копирования

Преимущества:

- быстрое восстановление данных,
- простое управление,
- все данные содержатся в одной резервной копии.

Недостатки:

- требует много места для хранения резервных копий,
- высокая загрузка сети,
- длительное выполнение резервного копирования.

Инкрементные бэкапы (Incremental backup)

Инкрементное резервное копирование использует полную копию, как начальную точку.

Затем выполняется резервное **копирование только блоков данных, которые были изменены** с момента последнего резервного задания, с заданным периодом выполнения задания.

В зависимости от политики хранения резервных копий через определенный период создается новая полная копия для повторения цикла.

Преимущества и недостатки Incremental Backup

Преимущества:

- высокая скорость резервного копирования (копируются только блоки измененных данных),
- меньше места для хранения (по сравнению с полным),
- большее количество точек восстановления.

Преимущества и недостатки Incremental Backup

Недостатки:

- низкая скорость восстановления данных (необходимо восстановить как начальную полную копию, так и все последующие блоки);
- менее надежна (зависит от целостности всех блоков в цепочке).

Дифференциальный бэкап (differential backup)

Дифференциальный бэкап — это тип резервного копирования файлов, при котором копируются не все исходные файлы, а только новые и измененные с момента создания предыдущей полной копии.

Он является чем-то средним между полным резервным копированием и инкрементальным. Название этого типа произошло от английского слова **Differential backup** и является *накопительным*, т.е. каждая следующая копия содержит все новые/измененные файлы с момента создания предыдущей полной резервной копии.

Преимущества и недостатки дифференциального бэкапа

Преимущества:

- Относительно небольшой размер разностной резервной копии, по сравнению с полной;
- Скорость создания в разы выше, чем полного бэкапа;
- Для восстановления файлов потребуется последний созданный полный бэкап и последний дифференциальный.

Недостатки:

- Избыточность данных, так как дифференциальный бэкап является накопительным.



Холодное резервирование БД

Холодное резервное копирование выполняется на уровне операционной системы при остановленной базе данных — резервируются файлы, составляющие базу данных файлы данных, управляющие файлы, файл параметров.

Преимущества и недостатки холодного резервирования БД

Преимущества:

- быстрое восстановление данных,
- простое управление,
- все данные содержатся в одной резервной копии.

Преимущества и недостатки холодного резервирования БД

Недостатки:

- можно восстановить только то состояние базы данных, которое было в момент остановки;
- транзакции, сделанные после рестарта базы, в резервную копию не попадут;
- не у каждой базы данных есть технологическое окно, когда базу можно остановить.



Горячее резервирование БД

При горячем резервировании база данных включена и открыта для потребителей.

Копия базы данных приводится в согласованное состояние путем автоматического приложения к ней журналов резервирования по окончании копирования файлов данных.

Преимущества и недостатки горячего резервирования

Преимущества:

- База данных может использоваться во время резервного копирования;
- База данных может восстановиться до определенного момента времени;
- Высокая скорость восстановления;
- Почти все объекты базы данных могут быть восстановлены.

Преимущества и недостатки горячего резервирования

Недостатки:

- В момент начала копирования содержимое базы данных может не совпадать с содержимым файлов;
- Во время копирования содержимое базы может меняться;
- Поскольку запись данных в базу и чтение файлов БД никак не синхронизированы, программа резервного копирования может прочитать некорректную страницу, в которой половина будет от старой версии страницы, а другая половина – от новой.

Репликация

Идея репликации основана на том, что кроме «главного» сервера («Мастера») постоянно работают **ведомые** сервера («слейвы»), которые получают инкрементные бэкапы с мастера в режиме реального времени.




Восстановление данных

Восстановление данных

Восстановление данных — процесс восстановления базы данных до исходного состояния из имеющейся резервной копии.

В процессе обычно используются утилиты для восстановления, поставляемые в комплекте с СУБД.

Как правило, существует подробная документация, как проводить восстановление.



Резервное копирование и восстановление PostgreSQL

Резервное копирование и восстановление PostgreSQL

- Выгрузка в SQL;
- Резервное копирование на уровне файлов;
- Непрерывное архивирование и восстановление.



Выгрузка в SQL

Идея, стоящая за этим методом, заключается в генерации текстового файла с командами SQL, которые при выполнении на сервере пересоздадут базу данных в том же самом состоянии, в котором она была на момент выгрузки.

[SQL Dump](#)



Резервное копирование на уровне файлов

Альтернативной стратегией резервного копирования является непосредственное копирование файлов, в которых Postgres хранит содержимое базы данных;

Непрерывное архивирование и восстановление

В процессе работы Postgres ведёт **журнал предзаписи** (WAL). В этот журнал записываются все изменения, вносимые в файлы данных.


Если происходит крах, целостность СУБД может быть восстановлена в результате «воспроизведения» записей, зафиксированных после последней контрольной точки.



Непрерывное архивирование и восстановление

Однако наличие журнала делает возможным использование третьей стратегии копирования баз данных: можно сочетать резервное копирование на уровне файловой системы с копированием файлов WAL.

Если потребуется восстановить данные, мы можем восстановить копию файлов, а затем воспроизвести журнал из скопированных файлов WAL, и таким образом привести систему в нужное состояние.



Резервное копирование и восстановление MySQL

Резервное копирование и восстановление MySQL

- Копирование файлов базы;
- Копирование через текстовые файлы;
- Инкрементные бэкапы;
- Репликация.

Копирование файлов базы

Базу данных MySQL можно скопировать, если временно выключить MySQL-сервер и просто скопировать файлы из папки */var/lib/mysql/db/*.

Если сервер не выключить, вероятна потеря и порча данных. Для больших нагруженных баз эта вероятность близка к 100%.

При первом запуске с «грязной» копией базы данных MySQL-сервер начнет процесс проверки всей базы, который может затянуться на часы.

Копирование файлов базы

В большинстве «живых» проектов регулярное выключение сервера БД на длительное время неприемлемо. Для решения этой проблемы применяется трюк, основанный на снимках файловой системы.

Общая схема действий такова:

- блокируются все таблицы,
- сбрасывается файловый кэш БД,
- делается снимок файловой системы,
- разблокируются таблицы.



Копирование файлов базы

Далее файлы спокойно копируются из снимка, который затем уничтожается. «Блокирующая» часть такого процесса занимает время порядка нескольких секунд, что уже терпимо.

В качестве расплаты на какое-то время, пока «жив» снимок, снижается производительность файловых операций, что в первую очередь бьет по скорости операций записи в базу.



Копирование через текстовые файлы

Для того, чтобы считать в бэкап данные из production-базы, не обязательно дергать файлы. Можно выбрать данные запросом и сохранить их в текстовый файл.

Для этого используется SQL-команда `SELECT INTO OUTFILE` и парная ей — `LOAD DATA INFILE`.



Копирование через текстовые файлы

Выгрузка производится построчно (можно отобрать для сохранения только нужные строки, как в обычном `SELECT`). Структура таблиц нигде не указывается — об этом должен заботиться программист. Он также должен позаботиться о включении команд `SELECT INTO OUTFILE` в транзакцию, если это необходимо для обеспечения целостности данных.

На практике `SELECT INTO OUTFILE` используется для частичного бэкапа очень больших таблиц, которые нельзя скопировать никаким другим образом.



Инкрементные бэкапы

Традиционно рекомендуют держать 10 бэкапов: по одному на каждый день недели, а также бэкапы двухнедельной, месячной и квартальной давности — это позволит достаточно глубоко откатиться в случае порчи каких-либо данных.

Храниться бэкапы должны точно *не на том же диске*, что и живая база, и не на том же сервере.



Инкрементные бэкапы

Эти требования могут стать проблемой для больших баз.

Частично решить эту проблему позволяют инкрементные бэкапы, когда полный бэкап делается, скажем, только по воскресеньям, а в остальные дни пишутся только данные, добавленные или измененные за прошедшие сутки.

Репликация

Реплика — это полная копия базы, но это **не резервная копия!**


Идея репликации основана на том, что кроме «главного» сервера («Мастера») постоянно работают ведомые сервера MySQL («слейвы»), которые получают инкрементные бэкапы с мастера в режиме реального времени..



Репликация

Таким образом, время отката уменьшается почти до сетевого лага. В случае краха Мастера можно оперативно назначить «новым Мастером» один из слейвов и перенаправить клиентов на него.

Кроме того, слейвы могут обрабатывать запросы на чтение данных (SELECT-ы); это можно использовать для выполнения каких-то расчетов или снижения нагрузки на мастера.



Мониторинг состояния бэкапов

Мониторинг состояния бэкапов

После создания любых типов бэкапов обязательно следить за их состоянием.

Важные аспекты, на которые следует обратить внимание:

- восстанавливаемость,
- целостность файлов и цепочки (в случае инкрементальных бэкапов),
- свободное место для хранения.

Удобным инструментом для такого мониторинга является Zabbix.



Итоги

Итоги

Сегодня мы:

- узнали, что такое резервное копирование БД;
- рассмотрели типы бэкапов, и что такое восстановление данных;
- ознакомились с резервным копированием в PostgreSQL и MySQL;
- узнали, на что обратить внимание при уже созданных бэкапах.



Домашнее задание

Домашнее задание

Давайте посмотрим ваше [домашнее задание](#).

- Вопросы по домашней работе задавайте **в чате** мессенджера Slack.
- Задачи можно сдавать **по частям**.
- Зачёт по домашней работе проставляется после того, как **приняты все задачи**.

**Задавайте вопросы и
пишите отзыв о лекции!**

Роман Гордиенко