



Oracle

DATABASE

История

► Основатели:

Ларри Эллисон

Боб Майнер

Эд Оутс

Главным преимуществом после выхода
была высокая скорость обработки
огромных массивов информации.



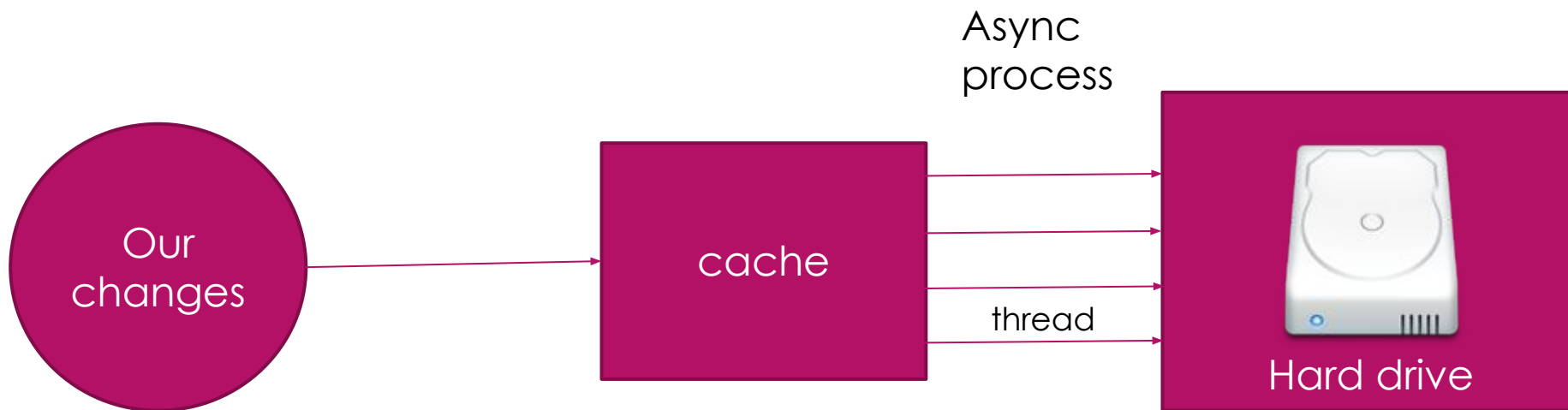
История

- ▶ Oracle стала исторически первой и одной из наиболее развитых реализаций архитектуры клиент/сервер.
- ▶ Еще одной важной особенностью системы стала полная реализация возможностей нового языка запросов SQL — подзапросы, операция соединения и так далее. Благодаря этому многократно выросла производительность труда SQL-программистов.
- ▶ В 1983 году на рынок вышла Oracle 3. Она была полностью переписана на C.



Ларри Эллисон — основатель Oracle

Что позволяет БД Oracle работать так быстро?



Механизм восстановления данных

- ▶ В СУБД Oracle можно включить архивацию оперативных журнальных файлов, и все изменения будут архивироваться. Таким образом при потере любого диска с блоками данных мы можем восстановить их на любой момент времени, включая момент прямо перед падением, накатив на последние архивные журнальные файлы текущий оперативный журнал.

Stand by копия

- ▶ Вышеупомянутые архивные файлы можно отправлять по сети и на лету применять к копии БД. Таким образом у вас всегда под рукой будет горячая копия с минимальным запаздыванием данных. В некоторых приложениях, где нет необходимости показывать данные с точностью до последнего момента, можно настроить такую БД только на чтение и разгрузить основной экземпляр БД, причем таких экземпляров на чтение может быть несколько.

Неблокирующее чтение и сегмент отката

- ▶ Одной из наиболее замечательных особенностей СУБД Oracle является неблокирующее чтение, которое достигается за счет сегмента отката. Запросы к Oracle на чтение **никогда** не блокируются, так как данные почти всегда могут быть прочитаны из сегмента отката.

Уровни изоляции транзакций

- ▶ В Oracle вообще нет уровня изоляции READ_UNCOMMITTED. Дело в том, что в других базах данных он используется для достижения максимального параллелизма путем удаления блокировок чтения. Но в Oracle чтение и так всегда выполняется без блокировок, таким образом мы уже имеем все преимущества, которые может дать этот уровень, не вводя никаких дополнительных ограничений.

Позвольте Oracle кэшировать ваши данные эффективно

- ▶ В Oracle все данные читаются-пишутся не прямо на диск, а через кэш. По умолчанию кэш основан на LRU алгоритме, так что если вы читаете какую-нибудь очень большую табличку по идентификатору в больших количествах, запрашивая в каждый раз новую строчку, то такие запросы могут вытеснять из кэша небольшую статическую табличку, которой бы самое милое дело постоянно находиться в кэше. Для таких целей при создании таблицы вы можете указать специальный вид кэша, куда будут ходить запросы к вашим таблицам. Так для первой таблицы в вышеописанном примере подойдет кэш RECYCLE, который по сути не хранит никакие данные, а сразу их выбрасывает из кэша. А для второй таблицы подойдет кэш KEEP, который позволить хранить в кэше небольшие статические таблицы и запросы ко всем остальным таблицам не будут вытеснять данные статических таблиц из кэша.

Проблемы

- ▶ Подвисание запросов на запись – причина в том, что журнальные файлы слишком большие или их слишком мало, а значит защищаемые ими данные не успевают сбрасываться на диск.

Пустые строки

- ▶ В Oracle есть одна очень интересная особенность, от которой они теперь уже никогда не смогут избавиться. Дело в том, что если вы кладете в БД пустую строку, то она сохранится как NULL. Таким образом при последующем чтении вы никогда не получите пустой строки, а только NULL. Имейте так же в виду, что по этой же причине пустые строки не попадают в индекс, так что если вы будете делать запросы, план выполнения которых, будет использовать индекс, то ваше пустые (вернее NULL) строки вы никогда не получите, но об этом чуть позже.