

Гатилин Дмитрий Николаевич,
студент магистратуры 2 курса гр. ИСТмз-31,
ФГОБУ ВО «Поволжский государственный университет
телекоммуникаций и информатики»
Gatilin Dmitrii Niloaevich,
2st year master's student, gr. ISTmz-31,
FGOBU in «Volga State University
of Telecommunications and Informatics»

Козлов Вячеслав Васильевич, к.т.н, доцент,
доцент кафедры информатики и вычислительной техники
ФГОБУ ВО «Поволжский государственный университет
телекоммуникаций и информатики»,
Самарский государственный технический университет (СамГТУ)
Kozlov Vyacheslav Vasilevich, k.t., associate,
Associate Professor of the Department of Computer Science,
FGOBU in «Volga State University
of Telecommunications and Informatics»,
Samara State Technical University (SamGTU)

**МЕТОДОЛОГИЯ ПОСТРОЕНИЯ ЭФФЕКТИВНЫХ
ЗАПРОСОВ ДЛЯ OLTP СИСТЕМ И OLAP ХРАНИЛИЩ
ДАННЫХ НА ПРИМЕРЕ СУБД ORACLE
METHODOLOGY FOR BUILDING EFFICIENT QUERIES
FOR OLTP SYSTEMS AND OLAP DATA WAREHOUSES
USING THE ORACLE DBMS EXAMPLE**

Аннотация. В статье проводится сравнительный анализ архитектурных и функциональных особенностей систем оперативной обработки транзакций (OLTP) и оперативной аналитической обработки данных (OLAP) в контексте СУБД Oracle. Исследуются специфические требования к построению и оптимизации SQL-запросов для каждой категории систем, обусловленные фундаментальными различиями в характере нагрузки и целях обработки данных.

Abstract. This paper presents a comparative analysis of the architectural and functional features of Online Transaction Processing (OLTP) and Online Analytical Processing (OLAP) systems in the context of the Oracle DBMS. It investigates the specific requirements for constructing and optimizing SQL queries for each system category, driven by fundamental differences in the nature of the workload and data processing objectives.

Ключевые слова: OLTP-системы, OLAP-системы, оптимизация SQL-запросов, транзакционная обработка, аналитическая обработка, архитектура Oracle.

Keywords: OLTP systems, OLAP systems, SQL query optimization, Transactional processing, Analytical processing, Oracle architecture.

Современные информационные системы разделяются на две основные категории: OLTP (Online Transaction Processing) и OLAP (Online Analytical Processing). OLTP-системы ориентированы на обработку большого числа одновременных транзакций с высокой скоростью и минимальной задержкой. Системы OLAP, напротив, предназначены для комплексного анализа данных, поддержки принятия решений и требуют обработки больших



объемов информации с акцентом на сложные аналитические запросы. Обе категории имеют различные архитектурные и функциональные особенности, которые значительно влияют на подходы к построению эффективных SQL-запросов.

Обзор систем OLTP и их особенности

OLTP-системы предназначены для обеспечения быстрой и предсказуемой обработки транзакций в режиме реального времени, выступая основой ежедневных операций предприятий с интенсивным вводом, обновлением и удалением данных. В таких системах транзакции обычно охватывают небольшие объемы информации, но при этом обрабатываются в большом количестве одновременно, что требует высокой пропускной способности и минимальных задержек при выполнении запросов. OLTP охватывает приложения с плотным потоком транзакций и многопользовательским доступом, включая банковские системы, системы бронирования и ввода заказов.

Архитектурные особенности OLTP в Oracle включают использование нормализованных схем баз данных, минимизирующих избыточность и обеспечивающих целостность данных при частых изменениях. Ключевым элементом является транзакция – логическая единица работы, совершающаяся полностью или не совершающаяся вовсе, с соблюдением принципов ACID: атомарности, непротиворечивости, изолированности и долговечности. Эти свойства гарантируют корректное состояние данных даже при высокой нагрузке и параллельном выполнении множества транзакций.

Для Oracle OLTP-среды важными компонентами являются сегменты отката, позволяющие реализовать откат и восстановление состояния данных, а также индексные структуры, кластеры и хеширование, обеспечивающие быстрый доступ к данным. Однако чрезмерное их использование негативно сказывается на производительности операций вставки и обновления, что требует баланса между индексами и скоростью транзакций. Размер блока данных, динамическое выделение пространства и многопоточный сервер – дополнительные настройки, влияющие на пропускную способность и устойчивость системы под нагрузками.

Оптимизация SQL-запросов в OLTP-системах стратегически важна, поскольку затрагивает как скорость обработки отдельных транзакций, так и общее время отклика системы под многопользовательской нагрузкой. Oracle предоставляет средства мониторинга и оптимизации, включая автоматическое распределение ресурсов и планирование запросов. Внедрение Oracle Autonomous Transaction Processing (ATP) позволяет существенно повысить масштабируемость, безопасность и автоматизацию управления транзакционными нагрузками за счет встроенного самообучения и адаптивной оптимизации [3].

В совокупности, характерные для OLTP-систем Oracle параметры и архитектура направлены на поддержание высокого параллелизма и доступности при обеспечении согласованности данных в режиме реального времени. Такой подход требует специализированного построения запросов, учитывающих особенности транзакционной обработки, быстрое выполнение DML-операций и минимизацию блокировок. Необходимость оптимизации вызвана постоянным ростом числа одновременно выполняемых транзакций и желанием предотвратить снижение производительности под нагрузкой, что формирует основу для разработки методологии построения эффективных OLTP-запросов [6].

Особенности OLAP систем и их реализация

Аналитическая направленность OLAP-систем заключается в предоставлении возможностей комплексного многомерного анализа данных для поддержки управленческих решений и бизнес-аналитики. OLAP-запросы ориентированы на выполнение сложных операций агрегации, фильтрации и срезов данных по различным измерениям, таким как время, география, категории или каналы продаж. В основе технологии лежит понятие OLAP-куба – многомерной структуры, содержащей взаимосвязанные показатели, что позволяет



представлять данные более осмысленно с точки зрения предметной области, нежели традиционные линейные таблицы [2].

Oracle OLAP реализован как расширение стандартной реляционной базы данных Oracle, объединяя преимущества масштабируемости, безопасности и управляемости платформы с многоуровневым аналитическим функционалом. Это позволяет выполнять многомерные вычисления в режиме реального времени и получать доступ к OLAP-объектам через нативный SQL, что существенно упрощает интеграцию аналитики в разнообразные приложения и дашборды.

Одной из ключевых особенностей Oracle OLAP является поддержка сложных бизнес-правил и использование предметно-ориентированных измерений, что освобождает аналитиков от необходимости знания глубинных особенностей хранения данных. Методология хранения опирается на концепции измерений (dimensions), показателей (measures) и многомерных кубов (cubes), которые могут быть как физически материализованы, так и виртуализированы над реляционными моделями, что обеспечивает большую гибкость и эффективность обработки. Схемы «звезда» и «снежинка» широко применяются в построении аналитических хранилищ для оптимизации запросов и обеспечения быстрого доступа к данным [5].

Для работы с большими объемами данных Oracle OLAP активно использует такие механизмы, как материализованные представления и партиционирование. Материализованные представления позволяют предварительно вычислять и сохранять результаты сложных агрегатных запросов, существенно снижая время отклика при повторном обращении. Партиционирование разбивает большие таблицы или индексы на управляемые сегменты по критериям, таким как временные интервалы или географические регионы, что ускоряет выборку данных и улучшает производительность параллельной обработки. Эти технологии вместе с адаптивным планированием запросов обеспечивают возможность масштабировать аналитическую нагрузку и эффективно работать с растущими объемами данных.

Интеграция OLAP непосредственно в Oracle Database исключает необходимость обмена данными между реляционными и многомерными системами, уменьшая задержки и повышая надежность аналитических приложений. Высокая производительность достигается за счёт параллелизма, кэширования и оптимизации многомерных вычислений. В результате Oracle OLAP широко применяется для построения корпоративных информационно-аналитических систем, аналитических витрин и отчетов, где важны интерактивность, гибкость анализа и возможность быстрое изменение параметров среза данных [1].

Подводя итог, особенности OLAP-систем Oracle с аналитической ориентацией и технологиями масштабируемой обработки больших объемов данных задают основу для разработки специализированных методологий построения запросов. Эти методологии отличаются от подходов в OLTP, учитывая вычислительную сложность, структуру запросов и характер данных.

Методология оптимизации запросов в OLTP-системах

Высокая скорость отклика SQL-запросов является критическим фактором для эффективного функционирования OLTP-систем на базе Oracle, поскольку задержки в обработке транзакций напрямую влияют на качество пользовательского опыта и общую производительность приложений. Минимизация времени выполнения достигается за счет оптимизации структуры запросов и правильного выбора методов доступа к данным, что требует комплексного подхода к настройке базы данных и продуманному использованию доступных технологий.

Одним из ключевых элементов оптимизации является тщательная настройка параметров памяти и распределения ресурсов. Правильно сконфигурированная SGA (System Global Area) и PGA (Program Global Area) обеспечивают эффективное кэширование данных и



выполнение операций сортировки без обращения к диску, что напрямую ускоряет выполнение транзакций. Важна также настройка параметров ввода-вывода и сетевых протоколов для снижения задержек на коммуникационном уровне [1].

Выбор и конфигурация индексов в OLTP-среде требуют особого внимания. Индексы B-Tree остаются наиболее распространенным инструментом для быстрого точечного поиска строк, используемых в коротких транзакциях. Для обеспечения масштабируемости и снижения затрат на обслуживание индексов целесообразно рассматривать их партиционирование при работе с большими объемами данных. Это позволяет локализовать операции обновления и ускоряет доступ, уменьшая конкуренцию за ресурсы [1].

Важной инновацией Oracle начиная с версии 11g стала OLTP-компрессия и компрессия неструктурированных данных (LOB Compression), которые позволяют существенно снижать объемы хранимых данных, не ухудшая производительность доступа. Такие методы уменьшают нагрузку на дисковую подсистему и облегчают управление хранилищем, что положительно влияет на время отклика.

Использование Result Cache – технологии кэширования результатов выполнения запросов и подзапросов – способствует сокращению количества повторных обращений к диску и убыстряет выполнение типовых выборок. Это особенно эффективно для часто вызываемых операций с неизменными данными в пределах сеанса работы пользователя.

Рекомендуется внедрение кэширования на уровне приложений и использование гетерогенных реплик для проведения асинхронных обновлений, что снижает нагрузку на основное OLTP-хранилище и повышает его устойчивость и масштабируемость. Такой подход позволяет минимизировать время отклика и увеличить пропускную способность, сохраняя при этом консистентность данных [4].

Основные рекомендации по построению эффективных запросов в OLTP-системах Oracle включают:

- минимизировать сканирование таблиц, максимально используя индексы для выборки по ключам;
- ограничивать количество возвращаемых строк запросами с использованием WHERE и фильтрации;
- избегать сложных объединений и агрегатных функций в критичных по скорости транзакций;
- применять партиционирование для распределения нагрузки и ускорения обслуживания больших таблиц и индексов;
- включать кэширование результатов для повторяющихся запросов, если данные статичны в течение сессии;
- регулярно проводить мониторинг и анализ планов выполнения с помощью инструментов Oracle, например, SQL Plan Management и Automatic Workload Repository (AWR).

Эффективное сочетание этих практик обеспечивает устойчивое выполнение большого количества параллельных транзакций с низкой задержкой и высокой надежностью, что является основой производительности OLTP-систем Oracle.

Методология построения эффективных запросов для OLAP-хранилищ

Аналитическая нагрузка в OLAP-системах характеризуется выполнением сложных многомерных агрегатных запросов, направленных на выявление закономерностей и трендов в больших объемах накопленных данных. Запросы часто включают многоуровневую группировку, срезы по измерениям и использование аналитических функций, что приводит к высокой нагрузке на вычислительные ресурсы и вводу-выводу. В таких условиях критически важно проведение предварительной агрегации данных и оптимизация порядка выполнения операций для сокращения времени отклика.



Oracle OLAP предлагает богатый набор встроенных SQL-функций для поддержки анализа числовых и символьных данных, включающих стандартные арифметические и строковые операции (ABS, ROUND, CONCAT, SUBSTR и др.), что расширяет возможности построения гибких аналитических запросов непосредственно в СУБД без необходимости дополнительной обработки на стороне приложений [4]. Использование этих функций эффективно интегрируется с механизмами OLAP, поддерживая разнообразные трансформации и вычисления внутри запросов.

Одной из ключевых особенностей проектирования запросов и объектов в OLAP является необходимость правильного порядка перечисления измерений при построении OLAP-кубов это напрямую влияет на производительность агрегации и скорость вычислений. Неправильно заданный порядок может значительно увеличить время формирования агрегатов, что делает этап проектирования кубов и загрузки данных критически важным при организации хранилищ.

Технология предварительной агрегации данных служит фундаментом снижения вычислительной нагрузки на этапе анализа. Загрузка данных в OLAP-кубы должна предусматривать предподсчет часто используемых агрегатов и использование материализованных представлений, что минимизирует объемы данных, обрабатываемых на лету во время вызова аналитических SQL-запросов. Такая стратегия существенно повышает производительность и позволяет достичь интерактивных временных откликов даже при обработке сложных вычислений [4].

Важную роль в повышении эффективности выполнения агрегационных запросов играет анализ планов выполнения SQL. Поскольку SQL-оптимизатор выбирает планы на основе статистических данных и внутренних алгоритмов, его решения не всегда являются оптимальными для конкретной аналитической нагрузки. В таких случаях применение подсказок, корректировка порядка операции объединений и агрегаций может значительно улучшить производительность запросов.

Кроме того, сравнительные исследования оптимизационных методов в OLAP на примере Oracle и других СУБД показывают, что успешная оптимизация зависит от учета особенностей конкретной платформы и правильного выбора методов агрегации, индексации и партиционирования. Поскольку Oracle OLAP тесно интегрирован с реляционной моделью, это позволяет использовать такой функционал, как партиционирование таблиц и материализованные представления для более эффективного управления большими объемами аналитических данных.

Итогом является необходимость создания специфических структур хранения данных, оптимизированных под аналитические запросы. Без применения этих специализированных структур и методов предварительной агрегации построение эффективных OLAP-запросов в Oracle становится существенно сложнее и менее результативно.

Таким образом, построение эффективных OLAP-запросов в Oracle требует тщательного проектирования многомерных структур и агрегаций, поддерживаемых набором встроенных функций и инструментальных средств, а также постоянной технической поддержки и анализа планов выполнения для обеспечения устойчивой высокой производительности хранилищ данных.

Практические рекомендации по выбору методов оптимизации в зависимости от типа системы

Выбор методов оптимизации SQL-запросов напрямую зависит от характера нагрузки системы – транзакционной или аналитической. В OLTP-средах ключевой задачей является обеспечение минимального времени отклика для большого числа мелких транзакций с частыми операциями вставки, обновления и точечного чтения. Это обуславливает необходимость акцента на использование традиционных row-store структур, продуманном индексировании и поддержании целостности данных с минимальными накладными



издержками. В таких случаях оптимизация ориентирована на сокращение блокировок, уменьшение объема сканируемых данных и обеспечение устойчивого высокого параллелизма.

В OLAP-системах требования иные: запросы направлены на обработку значительных объемов исторических данных с использованием сложных агрегаций, многомерных срезов и аналитических функций. Здесь важна оптимизация нагрузки ввода-вывода и вычислительных ресурсов за счет предварительной агрегации, использования материализованных представлений и партиционирования. Также целесообразно применять возможности расширенного кэширования и, при возможности, использовать колонковые индексы или специализированные колоночные СУБД, дополняющие функционал Oracle, что значительно ускоряет выполнение аналитических запросов.

Следует учитывать, что техники оптимизации должны соответствовать архитектурным особенностям хранения данных: традиционные OLTP базы эффективно работают с строчным форматом, а аналитика выигрывает от колоночного хранения, позволяющего быстро выполнять выборку по нужным столбцам и уменьшать нагрузку на дисковую подсистему. В Oracle это реализуется через гибкое сочетание реляционных инструментов, функций OLAP и продвинутых механизмов партиционирования.

Практическая оптимизация требует также правильной настройки аппаратных ресурсов. OLTP-среды выигрывают от конфигурации с низкими задержками ввода-вывода и балансированием нагрузки между процессорами, а аналитические решения требуют больших объемов оперативной памяти и высокой пропускной способности дисковой подсистемы для эффективной обработки больших массивов данных.

Нельзя пренебрегать процессами регулярного сопровождения: сбор статистики, реорганизация таблиц и индексов, обновление планов выполнения и мониторинг производительности критичны для поддержки оптимальной работы систем в изменяющихся условиях эксплуатации.

Обобщая, адаптация методик построения и оптимизации SQL-запросов должна базироваться на анализе конкретных требований нагрузки: OLTP – на быструю точечную обработку транзакций с минимальными накладными издержками, OLAP – на эффективную обработку масштабных аналитических вычислений с максимальным использованием предварительных агрегаций и специализированных индексов. Следующее практическое рассмотрение примеров позволит продемонстрировать применение этих рекомендаций в рабочих сценариях Oracle и иллюстрировать достижение оптимальной производительности в обоих типах систем.

Примеры оптимизированных SQL-запросов для OLTP и OLAP

Практическое применение методологий оптимизации запросов в Oracle для OLTP и OLAP-систем критично для достижения высокой производительности и надежности приложений. Рассмотрим примеры, демонстрирующие использование предложенных подходов на практике.

Для OLTP-систем характерны запросы с быстрым и точечным доступом к данным. Оптимизированный запрос для выборки активных сотрудников конкретного отдела выглядит так (Рисунок 1).

```
SELECT employee_id, first_name, last_name
FROM employees
WHERE department_id = :dept_id
AND status = 'ACTIVE';
```

Рисунок 1. Оптимизированный запрос



В данном запросе использование параметра:dept_id позволяет повторно использовать план выполнения, уменьшая затраты на парсинг. Индексы по столбцам department_id и status обеспечивают быстрый поиск, а фильтрация по ним минимизирует объем сканируемых данных. Такой запрос избегает сложных объединений и агрегатных функций, что снижает время отклика и блокировки в системе.

В OLAP-системах запросы, напротив, ориентированы на сложные агрегации и многомерный анализ. Пример оптимизированного аналитического запроса с использованием функций группировки (Рисунок 2).

```
SELECT department_id,  
       ROLLUP(month, product_category) AS grouping_set,  
       SUM(sales_amount) AS total_sales  
FROM sales_fact  
WHERE sale_date BETWEEN TO_DATE('2024-01-01','YYYY-MM-DD')  
AND TO_DATE('2024-06-30','YYYY-MM-DD')  
GROUP BY ROLLUP(month, product_category)  
ORDER BY department_id, month, product_category;
```

Рисунок 2. Пример оптимизированного аналитического запроса

В этом запросе оператор ROLLUP упрощает построение иерархических агрегатов по месяцам и категориям продуктов, что сокращает объем ручной работы при написании множества схожих запросов и ускоряет выполнение за счет использования встроенных возможностей Oracle. Предварительные агрегаты могут быть дополнены материализованными представлениями и партиционированием таблиц, что повышает производительность и снижает нагрузку на систему в период интенсивных аналитических сессий [2].

Использование таких методик – параметризация и индексация для OLTP, а также агрегирующие функции, материализованные представления и эффективное хранение данных для OLAP – позволяет не только повысить скорость обработки запросов, но и обеспечить устойчивую работу корпоративных систем при росте объемов данных и пользователей.

Таким образом, применение разработанных методик на практике демонстрирует значительный рост производительности баз данных Oracle, позволяя оптимально использовать ресурсы и поддерживать баланс между скоростью обработки транзакций и эффективностью аналитического анализа.

Список литературы:

1. How to Configure Oracle Database for OLTP Systems [Электронный ресурс] // philipmcclarence.com – Режим доступа: <https://philipmcclarence.com/how-to-configure-oracle-database-for-oltp-systems/>, свободный. – Загл. с экрана
2. OLAP-кубы – вчерашний день? Технологии нового / Хабр [Электронный ресурс] // habr.com – Режим доступа: <https://habr.com/ru/companies/neoflex/articles/910932/>, свободный. – Загл. с экрана
3. Oracle Autonomous Transaction Processing for Advanced OLTP [Электронный ресурс] // www.infolob.com – Режим доступа: <https://www.infolob.com/oracle-autonomous-transaction-processing-for-advanced-oltp/>, свободный. – Загл. с экрана
4. Оптимизация производительности БД Oracle [Электронный ресурс] // www.softlab.ru – Режим доступа: <https://www.softlab.ru/blog/oracle-optimization/>, свободный. – Загл. с экрана



5. Построение аналитической части корпоративной информационно-аналитической системы средствами Oracle OLAP Option и BI Beans [Электронный ресурс] // www.interface.ru
– Режим доступа: <https://www.interface.ru/home.asp?artid=9604>, свободный. – Загл. с экрана
6. Что для OLTP хорошо, то для OLAP – смерть / Хабр [Электронный ресурс] // habr.com
– Режим доступа: <https://habr.com/ru/companies/servermall/articles/948686/>, свободный. – Загл. с экрана

