

Проектирование Data Warehouse (DWH) - основы

Анализ источников

DWH способен одновременно собирать и анализировать данные из множества разнообразных источников. Эти источники могут быть чрезвычайно разнообразными, от внутренних корпоративных систем до внешних баз данных, FTP-серверов, а также socket-соединений и многих других. Это многообразие источников открывает широкие возможности для анализа и обработки информации.

В процессе интеграции в DWH данные из различных источников систематизируются и укладываются по строго определенным слоям. Эти слои можно сравнить с упорядоченными полками в библиотеке, где каждая книга (или в нашем случае, каждый фрагмент данных) имеет свое место. Например, в традиционной базе данных эти слои могли бы быть представлены в виде таблиц фактов, каждая из которых структурирована и организована для определенных целей.

Чтобы эффективно организовать данные в DWH, первостепенное значение имеет глубокое понимание характеристик и особенностей исходных данных. Важно учитывать такие факторы, как ограничения внешних ключей, наличие и использование индексов, функций, триггеров и хранимых процедур в исходных системах. Все эти аспекты напрямую влияют на то, как будет осуществляться процесс извлечения, преобразования и загрузки данных (ETL), что, в свою очередь, определяет эффективность и производительность всего хранилища данных.

Понимание модели данных в SQL источниках

1. Структура и схемы: В SQL-базах данные организованы в таблицы с четко определенной структурой. Понимание схемы таблиц, типов данных и их организации поможет в эффективном проектировании слоев DWH.
2. Отношения данных: В SQL важно понимание отношений между таблицами, включая первичные и внешние ключи. Это помогает в определении зависимостей и обеспечении целостности данных при их переносе в DWH.
3. Индексы и оптимизация: Индексы в SQL-базах ускоряют доступ к данным. При проектировании DWH важно учитывать, как индексация в источнике может повлиять на процессы ETL и производительность хранилища.
4. Триггеры и хранимые процедуры: Они часто используются для автоматизации операций в SQL. Понимание их роли в источниках данных поможет предотвратить потенциальные проблемы при миграции данных.

Понимание модели данных в NoSQL источниках

1. Гибкость схемы: В отличие от SQL, NoSQL базы часто имеют гибкую или динамическую схему. Это позволяет хранить разнородные данные, но требует особого внимания к структурированию данных при их переносе в DWH.
2. Типы NoSQL: Понимание различных типов NoSQL баз (документо-ориентированные, колоночные, ключ-значение, графовые) помогает определить оптимальные способы интеграции и агрегации данных в DWH.
3. Масштабируемость и распределение: NoSQL базы обычно хорошо масштабируются и поддерживают распределенное хранение. Это важно учитывать при проектировании архитектуры DWH и планировании ресурсов.
4. Консистентность данных: В NoSQL базах консистентность данных может быть настроена по-разному. Понимание принципов консистентности в источниках данных поможет в обеспечении точности и надежности данных в DWH.