DataMart-Optimierung

8 Tipps zur Performance-Verbesserung

DataMarts sind Bildausschnitte eines *Data Warehouses* und beinhalten Teildatenmengen (Kundendaten, Rechnungen, Umsätze, etc.) eines Datenbestandes. Eine Zunahme des Bestandes in einem DataMart führt häufig zu suboptimalen Prozessabläufen. Durch den damit verbundenen Geschwindigkeitsverlust von Anwendungen – oder im schlimmsten Fall Systemausfällen – gehen Unternehmen regelmäßig Ressourcen verloren.

Viele System-Administratoren wissen nicht, wo sie anfangen sollen, um den voran beschriebenen Performance-Problemen zu begegnen. Jede Situation benötigt ein individuell zugeschnittenes Verfahren. Wichtig ist, das Problem nicht auszusitzen, sondern aktiv zu handeln, andernfalls besteht die Gefahr, dass sich Probleme mit der Zeit potenzieren!

Zu diesem Zweck zeigt dieser Artikel acht mögliche Tipps zur DataMart-Optimierung auf.

BUSINESS.
DATA.
TECHNOLOGY.

CINTELLIC
CONSULTING GROUP

Generell wird zwischen **fachlichen und technischen Lösungsansätzen** zur Optimierung unterschieden:

Ein **fachlicher** Ansatz bedeutet, bestehende DataMart-Strukturen zu verändern und an neue Gegebenheiten anzupassen. Häufig ist dieser Schritt mit einem erheblichen Arbeitsaufwand verbunden. Falls richtig angewendet, kann dies allerdings einen signifikanten Performanceschub geben. Die **Tipps eins bis drei** sind häufig gewählte Ansätze zur fachlichen Performance-Optimierung.

Tipps vier bis acht sind **technische** Ansätze, bei denen versucht wird, durch Änderungen am System oder den Beladungs-ETLs Verbesserungen herbeizuführen. Technische Ansätze sind zumeist einfacher zu implementieren, haben allerdings auch weniger signifikante Auswirkungen auf die Performance im Vergleich zu fachlichen Ansätzen.

1. Logisches Design

Das **logische Design** eines DataMarts ist die Umsetzung eines konzeptionellen Entwurfs. Im Grunde ist es eine Abbildung der realen Business-Prozesse und definiert die benötigten Informationen, sowie die Verbindungen zwischen den Tabellen eines DataMarts.
Schlechte logische Designs führen zu einem nicht-performanten physikalischen Design. Performance-Probleme entstehen in der Folge durch miteinander verbundene Datensätze, welche über multiple Tabellen gestreut wurden. Solch eine Streuung verursacht einen erhöhten Verbrauch von Serverressourcen, was wiederum zu erhöhten IT-Kosten führen kann.

Um ein logisches Design zu erstellen, das einen expliziten Business-Need unterstützt, werden folgende Schritte benötigt:

a. Entitäten identifizieren
Analysten erstellen ein logisches Datenmodell, indem sie eine Business Funktion/Aktivität genau untersuchen. Durch die Untersuchung wird ein „User View" kreiert, welcher kritische Informationen für die Business Aktivität/Funktion repräsentiert. Stellen Sie sich vor, sie kreieren einen Webshop. Um diesen Shop am Laufen zu halten, benötigen Sie Informationen zu Verkäufen, Kunden, Produkten, Warenlager und Auslieferungen. Dies sind Ihre Entitäten.

b. Hinzufügen von Attributen
Anschließend werden Dimensionen zu den Entitäten zugeordnet. Diese Dimensionen sind beschreibende oder ergänzende Details der Entitäten. Zum Beispiel müssen Sie für die Entität ‚Kunde' wissen, welche Anschrift er besitzt, um ihm das bestellte Produkt auszuliefern. Zu den Verkäufen müssen Sie wissen, an welchem Datum ein Verkauf stattfand, welches Produkt verkauft wurde und wie hoch der Preis war.

c. Integration der verschiedenen Views
Zuletzt werden die gesammelten User Views in ein Gesamtbild integriert. Hierbei ist die Zuordnung der Beziehung zwischen Entitäten und Attributen wichtig.

2. Rücknahme der Normalisierung

Eine Normalisierung wird in Datenschemata vorgenommen, um Redundanzen zu vermeiden. Dies geschieht unter Einhaltung der Normalisierungsregeln für gewöhnlich, sobald Daten aus den Quellsystemen im Core Layer gespeichert werden. Für einen DataMart jedoch ist es in den meisten Fällen üblich, diese **Normalisierung aufzulösen**. Eine Denormalisierung ist unter anderem in den folgenden zwei Fällen gängige Praxis.

a. Aggregierte Tabellen
Aggregierte Tabellen speichern vorberechnete Resultate, welche über ein Set von Dimensionen zusammengefasst wurden (z. B. Summen- oder Maximierungsfunktionen). Dies ist eine beliebte Variante, um SQL-Abfragen und damit die Performance und den Datenspeicherbedarf zu optimieren.

b. Komplizierte Joins durch Vorberechnungen vermeiden
Falls häufig aufkommende DataMart-Abfragen auf komplexen Joins verschiedener Tabellen basieren, können die benötigten Daten in einer Tabelle gespeichert werden. Auf diese Weise werden die SQL-Abfragen beschleunigt, da Joins Rechenleistung kosten.

3. Historische Daten

Häufig beinhalten DataMarts viele **historisch gewachsene Datenbestände**, die im normalen Gebrauch obsolet sind.
Bei Abfragen auf den Tabellen ist die Abfragegeschwindigkeit spürbar langsamer. Als Gegenmaßnahme können selten benutzte historische Daten in Views oder eigens dafür bereitgestellte Tabellen verschoben werden. Falls historische Daten nicht benötigt werden, sollte die Möglichkeit des Löschens der Daten in Betracht gezogen werden.

4. Effiziente SQL Statements

Je nach Rahmenbedingungen können **SQL Statements effektiv oder ineffektiv** sein. Daher ist es wichtig den Kontext und das Ziel der zu optimierenden Abfragen zu definieren.
Damit nicht jede einzelne Abfrage eines DataMarts angepasst werden muss – was den Kosten-Nutzen-Rahmen negativ beeinflussen würde – ist es unerlässlich, vor Beginn der SQL-Anpassung zu eruieren, welche Abfragen häufig laufen, um dementsprechend fokussiert vorzugehen.

Beispielsweise sollte das **Kartesische Produkt vermieden** werden, wenn nicht genau das Ergebnis dessen gesucht wird. Das Kartesische Produkt ist dem Kreuzprodukt sehr ähnlich, dessen Resultat die Menge aller Kombinationen ist. Spezifisch bedeutet dies, dass jede Zeile einer Tabelle mit jeder Zeile einer anderen Tabelle kombiniert wird. Die Anzahl Zeilen gleicht der Multiplikation aller Zeilen der Ausgangstabellen. In der Praxis führt dies zu äußerst langsamen SQL-Abfragen.

Darüber hinaus ist es hilfreich die **Tabellengröße bei Abfragen zu reduzieren**. Dies kann durch geeignete Filter einfach mittels SQL umgesetzt werden. Auf diese Art und Weise werden nur Daten, welche von Interesse für die jeweiligen Prozesse sind, selektiert.

Da **Text-Strings viel Speicherplatz benötigen**, sind deren Abfragen langsam. Wenn möglich sollten nummerischen Werten daher auch nummerische Datentypen wie z.B. Integer für ganzzahlige Werte zugeordnet werden.
Viele Architekten vergessen in diesem Zusammenhang, die kleinstmögliche Datenstruktur zu verwenden. Je nach Datenbanksystem kann die Standardgröße von Datentypen variieren und dementsprechend viel Speicherplatz beanspruchen.

Ein weiteres hilfreiches Mittel gegen nicht-performante Abfragen sind **Sub-Selects**. Häufig ist dies ein Hilfsmittel, um komplexe Joins performanter zu gestalten. Dies sollte jedoch in jedem Fall getestet werden.

5. Indexierung

Ein Index ist ein Verweis, der eine Ordnungsrelation in einer Tabelle definiert. Ein Datenbanksystem kann anhand dieser Verweise die **Geschwindigkeit von Abfragen beschleunigen**. Die Erstellung eines Indexes ist eine redundante Datenspeicherung und damit – genau genommen – eine technische Denormalisierung.

Einige Datenbankmanagementsysteme sind in der Lage, einen Index automatisch zu kreieren und zu aktualisieren, wenn die Daten in der Basistabelle verändert werden. Für manuelle Herstellungen von Indizes kann der genaue Indexierungstyp mittels Prüfungstools bestimmt werden. **Indizes können eine Leistungssteigerung bei Lesezugriffen bewirken**, da eine gezielte Suche nach bestimmten Sätzen unterstützt wird. Oft werden Schreibzugriffe jedoch umso langsamer, je mehr Indizes vorhanden sind, da sowohl die Daten in der Tabelle, als auch die Daten in den Indizes aktualisiert werden müssen.
Falls eine Tabelle in den Speicher geladen werden kann, sollte jedoch von einer Indexierung absehen werden, da Speicherabfragen schneller sind.

6. Partitionieren

Da Indexierungen nicht immer das gewünschte Resultat erbringen, kann alternativ – oder zusätzlich – eine **Partitionierung** vorgenommen werden. Das Prinzip dahinter ist das Aufteilen einer großen Tabelle in mehrere kleine Tabellen. Je nach Aufbau der zu partitionierenden Tabelle(n) wird sich des horizontalen oder des vertikalen Partitionierens bedient.
Generell ist die Vorgehensweise nützlich, um große Tabellen mit einer Vielzahl an Zeilen performanter zu lesen oder zu schreiben. Damit sich die Partitionierung lohnt, sollte zunächst ein gewisses Datenvolumen vorhanden sein. In der Praxis beschränkt sich das folglich auf Faktendaten.
Die vertikale Partitionierung fragmentiert Tabellen anhand ihrer Spalten und speichert diese in einer neuen Tabelle. Die horizontale Partition hingegen fragmentiert Tabellen anhand ihrer Zeilen und speichert diese in einer neuen Tabelle. Bei beiden Herangehensweisen handelt es sich jedoch logisch und abfragetechnisch um ein und dieselbe Tabelle. Die erstellten Fragmente sind die Partitionen.
Um jedem Datensatz eine eindeutige Partition zuzuweisen, muss die Partitionsfunktion entsprechend konfiguriert werden.

7. Statistiken Up-To-Date halten

Ein vielmals vergessenes aber nicht zu unterschätzendes Hilfsmittel für performante DataMarts ist die **statistische Analyse**. Die Statistik sollte möglichst oft aktualisiert werden, denn z.B. Oracle's Cost-Based-Optimizer (CBO) benutzt Statistiken um sich für einen Ausführungsplan von Abfragen zu entscheiden. Wenn die Statistiken nicht korrekt sind, sind die Entscheidungen des CBO's fehlerhaft und führen damit zu einer verschlechterung der Performance. Je nach Datenbanksystem gibt es unterschiedliche Herangehensweisen um die Statistiken zu aktualisieren.

8. Beladung eines DataMarts

Da DataMarts aus dem Core Layer beladen werden, was performance-intensiv ist, sollte die **Beladungsfrequenz durchdacht** sein. Je nach Verwendungszweck der Daten muss dies in near-time oder in zyklischen Abständen, z. B. einmal am Tag passieren. Die korrekte Beladungszeit ist ebenfalls zu bedenken. Meist bietet sich hierfür die Nacht an, da die Datenbanksysteme in den nächtlichen Stunden kaum ausgelastet sind.
Falls die benötigten Daten lediglich für einen Report zu Beginn des Tages genutzt werden, bietet sich eine tägliche Aktualisierung in der Nacht an. Da auch die Beladungsart auf die Performance einwirkt, sollte diesem Punkt ebenfalls Aufmerksamkeit geschenkt werden.
Bei großen Datenbeständen eignet sich zumeist das **Delta-Verfahren**, bei dem nur die neu hinzugekommenen bzw. geänderten Datensätze geladen werden. Bei kleinen oder sich verändernden Datenbeständen ist ein **Full-Verfahren** zu empfehlen, bei dem immer alle Datensätze der Quelle geladen werden.

Zusammenfassung

Die Zunahme von Datenbeständen innerhalb eines DataMarts führt häufig zu nicht-performanten Prozessen. Dadurch verlangsamen sich Anwendungen. Es kann zu Systemausfällen und Ressourcenverlusten führen.
Da ein Aussitzen des Problems zu einer Potenzierung dessen führen würde, sollten sich Datamart-Verantwortliche frühstmöglich um Optimierungsmaßnahmen kümmern. Die acht genannten Tipps bilden einen guten Überblick über mögliche Verbesserungen.
CINTELLIC unterstützt Sie gern bei der Gestaltung einer nachhaltigen und performanten DataMart-Architektur.

von Stefan Neubert,
CINTELLIC Consulting Group

Ansprechpartner

Dr. Jörg Reinnarth
Geschäftsführer
CINTELLIC Consulting Group
joerg.reinnarth@cintellic.com

Stephan Klöckner
Senior Manager
CINTELLIC Consulting Group
stephan.kloeckner@cintellic.com

Cintellic im Social Web

Cintellic GmbH
Remigiusstraße 16
53111 Bonn
t +49 228 92 18 20
info@cintellic.com
www.cintellic.com

Über CINTELLIC

Die 2010 gegründete CINTELLIC Consulting Group ist eine auf digitales Kundenmanagement spezialisierte Unternehmensberatung, die ihre Klienten vom ersten Konzept bis zur Umsetzung in der Praxis ganzheitlich begleitet. An den Standorten in Bonn, Frankfurt am Main und München arbeiten rund 60 Mitarbeiterinnen und Mitarbeiter.

Zu den Klienten zählen DAX-Konzerne, führende mittelständische Unternehmen und insbesondere zahlreiche sogenannte „Hidden Champions" mit den Branchenschwerpunkten Banken und Versicherungen, Telekommunikation, IT, Medien, Unterhaltung, Handel, E-Commerce, Versorger und Logistik.

www.cintellic.com

#jointheteam

CINTELLIC befindet sich auf Wachstumskurs. Vielleicht mit Ihnen? Jetzt Stellenanzeigen entdecken und bewerben!
https://www.cintellic.com/stellenangebote/

CINTELLIC
CONSULTING GROUP