

**Minimal-invasives Performance-Tuning
bei komplexen Anfragen**

Prof. Dr. Wolfgang Benn

**TU Chemnitz
Datenverwaltungssysteme
09107 Chemnitz**

**dimensio informatics GmbH
Brückenstr. 4
09111 Chemnitz**

1. Einführung

In den letzten 10 bis 15 Jahren hat sich der Bereich *Business Intelligence Anwendungen*, bei denen Datensätze aus unterschiedlichen Datenbanken zusammengeführt werden, um unternehmenspolitische Zusammenhänge darin zu erkunden bzw. strategische Informationen daraus zu gewinnen, fest im Markt etabliert. Indexe über den Datenbeständen sind für solche Applikationen zwar eine Selbstverständlichkeit, verhindern jedoch nicht, dass Anfragen zum Teil Minuten, Stunden oder gar Tage dauern, ehe die gewünschten Antworten erzielt werden können.

Ein Grund dafür findet sich in der traditionellen Vorgehensweise, einen Index zu erstellen. Üblicherweise wird dieser nämlich von außen – durch den Menschen – nach mathematischen oder technischen Gegebenheiten vorgegeben. Inhaltliche Beziehungen zwischen den Datensätzen werden nur insoweit beachtet, als der Mensch für sich in Anspruch nimmt, diese zu erkennen und sie – wenn überhaupt möglich – in die Definition des Indexaufbaus zu integrieren. Ob das jedoch immer zu einem optimalen Ergebnis führt, ist nicht belegt.

Ein einfaches Alltagsbeispiel soll hier kurz das Problem der semantischen Beziehungen zwischen Suchraumdimensionen charakterisieren: In jedem Kochbuch befinden sich Rezepte für unterschiedliche Arten von Speisen, z.B. Mehlspeisen, Suppen, Speisen mit Fleisch, etc. und jedem Rezept ist eine Liste mit Zutaten beigefügt.

Leidet nun eine Person unter Zöliakie (einer Unverträglichkeit gegenüber Getreideeiweiß und, damit meist einhergehend, auch gegenüber Milchzucker), ist sie darauf angewiesen, jedes Rezept einzeln bezüglich der unverträglichen Inhaltsstoffe zu begutachten, denn die normale Rezeptklassifizierung ist in diesem Fall untauglich.

Befinden sich die Rezepte jedoch in einer Datenbank, über der ein mehrdimensionaler Index für Zutaten und deren semantische Beziehung zueinander existiert – also das gemeinsame Auftreten oder gemeinschaftliche Fehlen von Zutaten im Index abgebildet ist – können Anfragen zur gängigen Rezeptklassifizierung der Rezepte ebenso abgegeben werden, wie solche, die bestimmte Krankheitsbilder berücksichtigen.

Der von *dimensio informatics* angebotene Datenbankindex ***dimensio*** realisiert einen solchen Index. Er bietet eine kompakte Datenstruktur in Gemeinschaft mit der Eigenschaft, die Daten in einem Datenbestand semantisch klassifizieren zu können. Er ist damit ein Ersatz für viele Sekundär- und/oder mehrdimensionale Spezialindexe. ***dimensio*** unterscheidet sich von gewöhnlichen Datenbankindizes durch seine selbstlernende Klassifikationskomponente, die den Datenbestand analysiert und die für einen Index notwendigen Intervallgrenzen selbstlernend und ohne Begrenzung der Attributanzahl (Dimensionen eines Suchraumes) aus dem Datenbestand heraus bestimmt.

Im Folgenden werden zunächst an einem Beispiel die Vorteile von ***dimensio*** für BI-Probleme an Hand eines Proof of Concept mit einer Oracle Exadata Quarter Rack Appliance über dem TPC-H Schema dargestellt. Abschließend wird auf die kostenminimierende minimal-invasive Integration des Index in bestehende IT-Landschaften eingegangen.

2. dimensio & BI

dimensio ist ein mehrdimensionaler semantischer Index für Anwendungsbereiche mit hochdimensionalen Datenbeständen – d.h. Bereiche mit komplexen Anfragen, in denen sehr viele Attribute zu korrelieren sind, um eine Antwort zu erhalten.

Der Index selbst ist ein eigenständiges, datenbankunabhängiges Produkt und besteht aus einer Klassifikations- und einer Verwaltungskomponente. Das Verfahren zur Klassifikation der Datenbankinhalte verwendet Künstliche Neuronale Netze und ist in der Lage, die Datensätze in einer Datenbank gemäß deren inhaltlichem Zusammenhang zu gruppieren. Anschließend wird diese Gruppierung in eine hoch effiziente Verwaltungsstruktur überführt und zur Indizierung verwendet.

Komplexe Anfragen finden sich u.a. im BI-Problembereich, wo in der Regel das gemeinschaftliche Vorhandensein bzw. die gemeinschaftliche Abwesenheit bestimmter Eigenschaften (Attributwerte) aus unterschiedlichen Basisrelationen nachgefragt wird. Wie das in der Praxis zu verstehen ist, zeigt sich am TPC-H Benchmark, der speziell für den Bereich *Decision Support* erstellt wurde und neben einem dementsprechenden Schema (s.u.) auch eine Anzahl von Nebenbedingungen und Anfragen bereithält, um technische Konfigurationen bezüglich ihrer Geschwindigkeit und ihrer Kosten pro Query gegeneinander auszutesten.

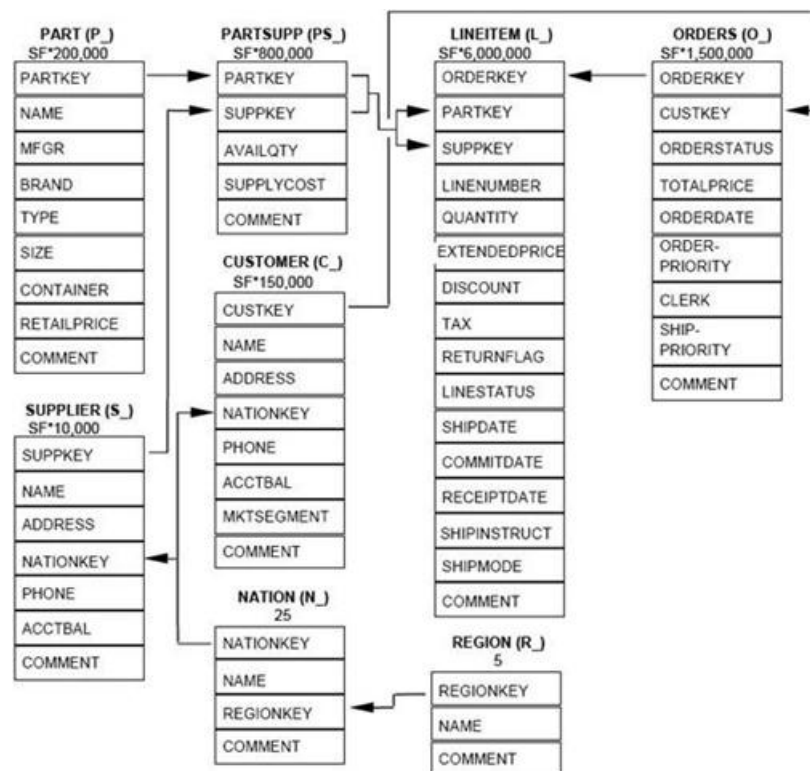


Abb. 1: TPC-H Schema

Der TPC-H Benchmark wird in unterschiedlichen Skalierungen durchgeführt, um den Anforderungen unterschiedlicher Unternehmensgrößen gerecht werden zu können. So besteht die kleinste Skalierungsstufe (TPC-H 1) aus 1 GB Nutzdaten, die Stufe drüber (TPC-H 10) aus 10 GB Nutzdaten, etc. etc.

Interessant für *dimensio informatics* war in diesem Kontext die Frage, ob der mehrdimensionale Index **dimensio** in einer High-Tech-Umgebung, wie der *Oracle EXADATA*, einen Geschwindigkeitszuwachs erzeugen kann – und wenn ja, wie groß dieser ausfallen würde.

3. BI-PoC: Exadata & *dimensio*

Die für den PoC notwendige Testumgebung wurde von der *ISE GmbH*, Nürnberg, einem Business-Partner der *dimensio informatics GmbH* zur Verfügung gestellt und bestand aus einer vorkonfigurierten Exadata Quarter Rack Appliance. Für den Test wurde das oben gezeigte Schema verwendet und neun Attribute (in Abb. 2 rot unterlegt), die als Selektionskriterium in einem inneren Verbund dienen können, durch **dimensio** indiziert.

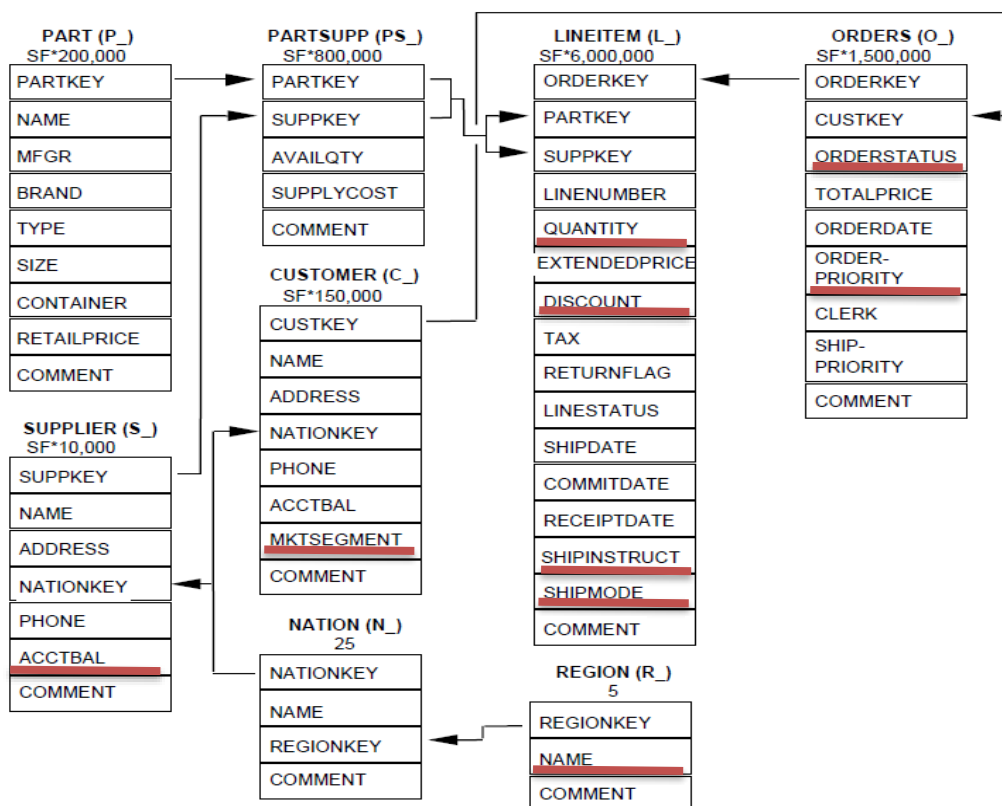


Abb. 2: Indizierte Felder im TPC-H-Schema

Im normalen Anfrageverlauf ohne **dimensio**, wird eine SQL-Anfrage direkt an das Datenbanksystem geschickt. Das DBMS wertet die Prädikatliste aus und korreliert die gefundenen Attributwerte - d.h. eine normale SQL-Anfrage folgt dem bekannten Muster:

```
select <Attributliste> from <Relationenliste> where <Prädikatliste>
```

welches wir im Folgenden als **M1** bzw. **Originalanfrage** bezeichnen werden. Eine – willkürlich aus der zum Test verwendeten Anfragemenge herausgegriffene - Beispielanfrage zeigt dieses Anfragemuster in der Praxis:

```

SELECT o.o_orderstatus,
       o.o_orderpriority,
       c.c_mktsegment,
       l.l_shipinstruct,
       l.l_shipmode,
       l.l_returnflag,
       l.l_discount,
       l.l_quantity,
       r.r_name,
       r2.r_name,
       s.s_acctbal
FROM   h_orders o,
       h_customer c,
       h_nation n,
       h_region r,
       h_lineitem l,
       h_part p,
       h_supplier s,
       h_nation n2,
       h_region r2
WHERE  o.o_custkey = c.c_custkey
       AND c.c_nationkey = n.n_nationkey
       AND n.n_regionkey = r.r_regionkey
       AND o.o_orderkey = l.l_orderkey
       AND l.l_partkey = p.p_partkey
       AND l.l_suppkey = s.s_suppkey
       AND s.s_nationkey = n2.n_nationkey
       AND n2.n_regionkey = r2.r_regionkey
       AND o.o_orderstatus in ('F')
       AND o.o_orderpriority in ('1-URGENT')
       AND c.c_mktsegment in ('HOUSEHOLD')
       AND l.l_shipinstruct in ('COLLECT COD')
       AND l.l_shipmode in ('FOB')
       AND l.l_returnflag in ('A')
       AND l.l_discount between 0 and 0.1
       AND l.l_quantity between 3 and 50
       AND r.r_name in ('EUROPE')
       AND r2.r_name in ('MIDDLE EAST')
       AND s.s_acctbal between 4693.26 and 4744.08;

```

Bei Einsatz von **dimensio** ergeben sich für das Anfragemuster zwei Alternativen. Die erste geht davon aus, dass der Index dem Datenbanksystem die Identifikatoren (PK) der gesuchten Datensätze zusätzlich zur Prädikatliste übergibt und damit die Verifikation der Antwortmenge dem Datenbanksystem überlässt. Diese Variante einer optimierten SQL-Anfrage folgt dem Muster

select <Attributliste> from <Relationenliste> where <Prädikatliste> and PK in <PK-Liste>

und wird als **M2**, bzw. als **optimiert PK erweitert** bezeichnet. Zur Verdeutlichung zeigen wir hier dieselbe Anfrage wie zuvor in dieser Form:

```

SELECT l.l_key,
       o.o_orderstatus,
       o.o_orderpriority,
       c.c_mktsegment,
       l.l_shipinstruct,
       l.l_shipmode,
       l.l_returnflag,
       l.l_discount,
       l.l_quantity,
       r.r_name,
       r2.r_name,
       s.s_acctbal
FROM   h_orders o,
       h_customer c,
       h_nation n,
       h_region r,
       h_lineitem l,
       h_part p,
       h_supplier s,
       h_nation n2,
       h_region r2

```

```

WHERE  o.o_custkey = c.c_custkey
      AND c.c_nationkey = n.n_nationkey
      AND n.n_regionkey = r.r_regionkey
      AND o.o_orderkey = l.l_orderkey
      AND l.l_partkey = p.p_partkey
      AND l.l_suppkey = s.s_suppkey
      AND s.s_nationkey = n2.n_nationkey
      AND n2.n_regionkey = r2.r_regionkey
      AND o.o_orderstatus in ('F')
      AND o.o_orderpriority in ('1-URGENT')
      AND c.c_mktsegment in ('HOUSEHOLD')
      AND l.l_shipinstruct in ('COLLECT COD')
      AND l.l_shipmode in ('FOB')
      AND l.l_returnflag in ('A')
      AND l.l_discount between 0 and 0.1
      AND l.l_quantity between 3 and 50
      AND r.r_name in ('EUROPE')
      AND r2.r_name in ('MIDDLE EAST')
      AND s.s_acctbal between 4693.26 and 4744.08
      AND L_KEY in (567611161, 555785090, 519781747, 499367351, 467113047,
396508704, 390164867, 368363409, 355328269, 353995232, 323517288, 311944817,
306210972, 285937194, 263122508, 247057804, 195807316, 166803060, 086605608,
083324744, 068114868, 060375778, 016944064, 590907671, 588563728, 581754607);

```

Eine zweite Alternative für die Optimierung mittels **dimensio** ist die Modifikation der Originalanfrage dahin gehend, dass dem Datenbanksystem ausschließlich die Identifikatoren der gesuchten Datensätze übergeben werden und die Verifikation der Antwortmenge durch das **dimensio**-System oder gar nicht erfolgt. Hier gilt das Anfragemuster

```
select <Attributliste> from <Relationenliste> where PK in <PK-Liste>
```

welches wir als **M3** oder **optimiert nur PK** bezeichnen. Allerdings wurde diese Form der Anfrage im TPC-H Test nicht weiter verwendet, da aus vorangegangenen Tests mit anderen Schemata bereits bekannt war, dass die Antwortzeiten konstant im unteren Millisekundenbereich liegen.

Um eine ausreichend große Realitätsnähe zu erzielen, wurden für den Test die TPC-H-Skalierungen 10 und 100 verwendet, d.h. der Speicherbedarf im Tablespace betrug für 10 GB Nutzdaten etwa 14 GB und für 100 GB Nutzdaten rund 152 GB.

Hierfür konnten die nachfolgenden Werte ermittelt werden:

TPC-H 10

Anfragemuster	Parallelität	Antwortzeit	I/O-Last
M 1	ohne Parallelitätszwang	17 Sekunden	
M 1	Parallelität 128 cores	2-3 Sekunden	ca. 2,1 GB/s
M 2	ohne Parallelitätszwang	0,03 - 0,2 Sekunden	kaum relevant

TPC-H 100

Anfragemuster	Parallelität	Antwortzeit	I/O-Last
M 1	ohne Parallelitätszwang	1:17 Minuten	
M 1	Parallelität 128 cores	8 - 19 Sekunden	ca. 3 GB/s
M 2	ohne Parallelitätszwang	0,2 - 1 Sekunde	kaum relevant

4. Minimal-invasive Integration

dimensio ist ein *minimal-invasives Tuningprodukt*, das – unabhängig vom Ort der Integration in eine bestehende IT-Landschaft – nur extrem geringen Aufwand verursacht. Um unterschiedlichen Kundenanforderungen entgegen zu kommen, gibt es technisch drei Möglichkeiten der Integration in eine bestehende IT-Infrastruktur bzw. Softwareanwendung.

Integrationspunkt 1 - die Anwendung

Dieser Ansatz kommt für Softwareanbieter (Softwarehäuser, OEMs) oder Unternehmen, die Zugriff auf den Quellcode ihrer Anwendung haben, in Frage, aber auch für sog. Endkunden. Änderungen am Code sind minimal, da in jedem Fall lediglich der **dimensio Client** seitens der Anwendung angesprochen werden muss und für diese Art der Integration eine eigene Programmbibliothek (API) ausgeliefert wird. Damit ist eine problemlose Integration rasch und kostengünstig möglich.

Für die Ausführung des **dimensio Servers** für Softwarehäuser und OEMs kann der Index vollständig und für Endkunden unsichtbar in eigene Anwendungen integriert werden. Softwarehäuser und/oder OEMs können auf diese Weise ihre hauseigenen Produkte extrem beschleunigen und quasi unter ihrem eigenen Namen von der Performancesteigerung profitieren.

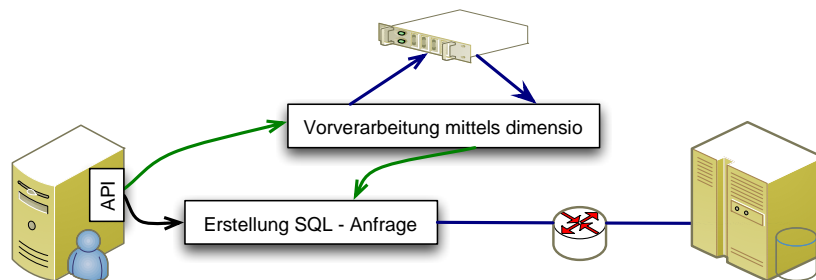


Abb. 3: Integration in die Anwendung

Wird zwar eine Anbindung aber keine Integration in die Anwendung gewünscht, was bei Endkundenprojekten vorkommen kann, kommen zunächst die **dimensio Virtual Appliance** und eventuell auch die **dimensio Appliance** als Lösung in Frage (Abb. 3).

Die **dimensio Virtual Appliance** läuft auf vom Kunden bereitgestellter IT, d.h. der Kunde stellt eine virtuelle Maschine mit der notwendigen Leistungsfähigkeit für den **dimensio Server** zur Verfügung und bietet auch den geforderten Speicherbereich (storage capacity). Ist das nicht möglich oder nicht gewünscht, kann die **dimensio Appliance** zum Einsatz kommen, denn diese wird von **dimensio informatics** komplett vorkonfiguriert und in einem eigenen Gehäuse mit allen notwendigen Anschlüssen ausgestattet ausgeliefert.

In beiden letzteren Fällen muss in der Anwendung lediglich das Ansprechen des **dimensio Client** organisiert werden, was in der Regel mit wenigen Codezeilen getan ist.

Integrationspunkt 2 - das Netzwerk

Eine sehr elegante Möglichkeit der Integration in eine bestehende IT-Infrastruktur ist für IT-Dienstleister, wie z.B. Rechenzentren und/oder Cloudbetreiber, die Integration in das Netzwerk (Abb. 4). Diese Integrationsvariante erfordert überhaupt keine Änderungen an der bestehenden IT-Infrastruktur. Weder die Anwendung, noch die Datenbank werden angefasst oder geändert.

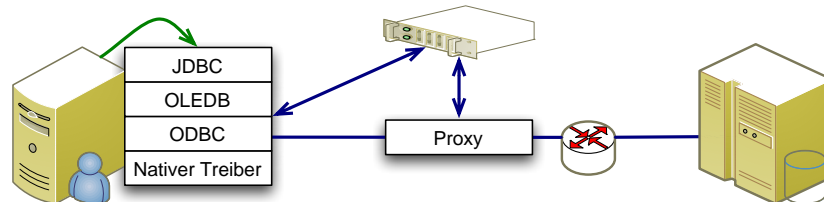


Abb. 4: Integration in das Netzwerk

Bei dieser Lösung wird der *dimensio Client* in den Datenbanktreiber integriert oder als Proxy ausgebildet. Der direkte Zugriff der Anwendung auf die Datenbank wird unterbunden, es wird lediglich eine Umleitung installiert, die sämtliche Anfragen an die Datenbank prüft. Der *dimensio Server* filtert dann die Anfragen: Anfragen über indizierte Tabellen werden an den Index weitergeleitet und dieser optimiert wie im Testbeispiel beschrieben. Anfragen über nicht indizierte Tabellen werden einfach durchgeroutet. Antworten, die von der Datenbank kommen, werden direkt an die anfragende Anwendung weitergereicht. Mit diesen Lösungen ist der Begriff *minimal-invasiv* am deutlichsten dokumentiert.

Integrationspunkt 3 - die Datenbank

Eine schnelle und einfache Integrationsvariante für Anwender, die Zugriff auf das Datenbanksystem haben oder auch für Dienstleister, die Datenbanksysteme anpassen, ist die Anbindung des *dimensio Servers* direkt an das Datenbankmanagementsystem (Abb. 5).

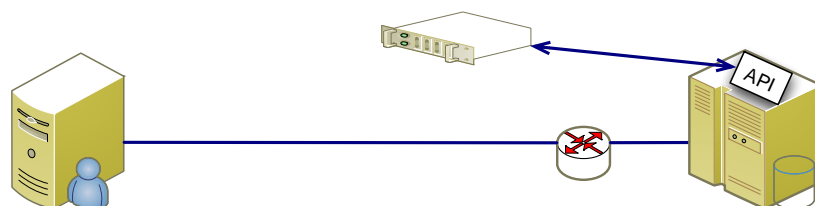


Abb. 5: Integration in das DBMS

Der *dimensio Client* wird bei dieser Integrationsvariante als Plug-In dem DBMS hinzugefügt. Dazu muss das DBMS eine entsprechende Schnittstelle bereitstellen. Derzeit bieten nur Oracle und DB2 Schnittstellen an, die einen für *dimensio* ausreichenden Funktionsumfang zur Verfügung stellen. Im Fall von Open-Source-Datenbanksystemen kann die Integration jedoch auch direkt im Code des Systems vorgenommen werden.

Erfolgt von Seiten der Anwendung ein Zugriff auf Tabellen die von *dimensio* indiziert sind, leitet das DBMS die Anfrage an die Schnittstelle und damit an den *dimensio Client* weiter. Der *dimensio Server* liefert dann auf Anfrage des Client die Indizierungsinformationen via Plug-In direkt an das DBMS.

5. Die Präsentation

In der Präsentation während der Konferenz (16:30 Uhr, Session: Data Warehouse & ETL) wird das Verständnis für den Performancegewinn des hier skizzierten Beispiels durch die Darstellung der Indexfunktionsweise vertieft. Weiterhin werden ein zweites Beispiel und eine TCO-Betrachtung für verschiedene Anwendungsbereiche – u.a. auch BI - gezeigt.

dimensio informatics GmbH - the spirit of speed®
Brückenstr. 4
09111 Chemnitz
Deutschland

Telefon +49 371 26 20 19 0
Telefax +49 371 26 20 19 10
info@dimensio-informatics.com
www.dimensio-informatics.com