Willkommen zum „IBM Informix Newsletter“

Inhaltsverzeichnis

Redaktion / Anmeldung

Liebe Leserinnen und Leser,
die vielen Anmeldungen zum Newsletter zeigen uns, dass das Thema INFORMIX auf großes Interesse trifft. Wir freuen uns über Ihr Interesse und hoffen, auch diesmal eine nützliche und interessante Zusammenstellung von Themen getroffen zu haben.
Die vorherige Ausgabe aus dem Oktober finden Sie unter „Aktuelles“ auf der Seite:
http://www-306.ibm.com/software/de/db2/informix/ids.html

Wenn Sie sich noch nicht für den Bezug des Newsletter angemeldet haben, dann sollten Sie dies möglichst bald tun um diesen Newsletter auch in Zukunft zu erhalten.
Hierzu senden Sie das Subject „ANMELDUNG“ an ifmxnews@de.ibm.com .
Wollen Sie keinen Newsletter mehr, dann senden Sie „ABMELDUNG“.

Haben Sie Fragen, Probleme, Bedarf an Schulungen oder Consulting zum Thema INFORMIX, dann nehmen Sie Kontakt mit uns auf: ifmxnews@de.ibm.com ,
wir betreuen Sie gerne.
Der Tipp zu DELIMIDENT wurde auf Grund der Anfrage eines Lesers des INFORMIX Newsletters erstellt (Vielen Dank nach Hamburg !).
Haben Sie auch Themenwünsche, dann schreiben Sie uns.

Kulturtipp:
Wenn Sie im Dezember eine Schulung in unserem Schulungszentrum in München besuchen, dann sollten Sie dies unbedingt auch mit einem abendlichen Besuch auf dem „Winter Tollwood“ verbinden.

Ihr TechTeam

TechTipp: Indexwahl

Entscheidend für die Performance einer Datenbank ist die Anlage der Indices.
Folgende Punkte können als Anhaltspunkte bei der Wahl von Indexen dienen:

- ❑ Das erste Feld im Index sollte möglichst selektiv sein (z.B. PLZ statt Land)
- ❑ Die Breite des Index (in Bytes) sollte maximal 1/3 der Breite der Datespalte betragen.
- ❑ Zusammengesetzte Indexe sind am effektivsten, wenn das erste Indexfeld das selektivste ist, die weiteren Felder in der Reihenfolge der Selektivität angehängt werden.
  Das ersten Feld eines Zusammengesetzen Index muss in der Abfrage vorkommen, damit der Index verwendet werden kann, weitere Felder können verwendet werden, wenn die Abfragekriterien alle Felder des Indexes bis zu diesem Feld abdecken.
- ❑ Zusätzliche Indexfelder zur Untersützung von Sortierung/Gruppierung an einen zusammengesetzten Index anzuhängen ist meist ratsam.
- ❑ Alle Bedingungen eines Tablejoin sollten wenn möglich im Index enthalten sein.

Der beste Index hilft allerdings nichts, wenn die Pflege der Datenbank mittels „update statistics“ und gezieltem „update statistics medium/high“ nicht erfolgt.
Die Statistik Informationen werden vom Optimizer verwendet, um eine Entscheidung über den optimalen Abfragepfad zu treffen. Damit der Optimizer richtig arbeiten kann, muss er über die richtigen Information über die Datenmengen und Datenverteilungen verfügen.

Findet der Datenbankserver keinen kostengünstigen Weg um die Daten mittels Index zu lesen, muss meist über einen sequential Scan die komplette Tabelle eingelesen und durchsucht werden.

Anzahl der Indices je Tabelle:
Global gilt, dass bei OLTP Systemen die häufigsten Abfragen auf große Tabellen durch Indices abgedeckt sein sollten.
Bei Datenbanken, auf die meist lesend zugegriffen wird, sind zusätzliche Indices sinnvoll, wenn dadurch sequentielle Scans auf den Daten vermieden werden können.
Bei sehr schreibintensiven Systemen sollte die Anzahl der Indices auf das notwendige Maß für Datenkonsistenz und die Hauptzugriffe beschränkt werden.

Mehr Infos hierzu im IBM Informix Dynamic Server Performance Guide:
Index Performance Considerations
(siehe Weblinks 1.)

TechTipp: Index Fragmentierung

Da die zu verwaltenden Datenmengen meist stetig steigen, werden auch die zugehörigen Indices zunehmend größer und können nicht mehr effizient im Memory gehalten werden.

Analog zur Fragmentierung von Tabellen bietet IDS die Möglichekt, auch Indices zu fragmentieren.
Die Indexfragmentierung hat folgende Vorteile:

- ❑ Die Instanz kann je Fragment entscheiden, ob der Teilindexbaum durchsucht werden muss oder nicht.
- ❑ Müssen mehrere Fragmente durchsucht werden, kann diese Suche parallel erfolgen.

Ab Version 10 kann statt der Fragmentierung in unterschiedliche Dbspaces auch eine Fragmentierung (Partitionierung) in einem Dbspace erfolgen. Da auch hier die Fragmente parallel durchsucht werden können, muss die Abbildung des Dbspaces auf die physikalischen Platten mitbedacht werden.

Die Fragmentierungsregeln sollten:

- ❑ Die Bedingungen der Abfrage abbilden
- ❑ Den Erfordernissen der Anwendung entsprechen (z.B. Produktionsdaten/Archivdaten)
- ❑ Die Parallelität in der Verarbeitung unterstützen

Indices können nicht nach „round robin“ fragmentiert werden.

Beispiel für eine Indexfragmentierung:

```
create index idate_order on orders(order_date, order_num)
     fragment by expressíon
     order_date < '01.01.2000'              in archivdbs
     order_date >= '01.01.2000'
          and order_date < '01.01.2006'     in datadbs2000
     oder_date >= '01.01.2006'              in datadbsakt
```

Um auch PrimaryKeys fragmentieren zu können, muss vor der Anlage des PrimaryKey Constraints ein fragmentierter Unique Index erstellt werden. Dieser wird dann bei Anlage des Constraints genutzt.

Mehr Infos hierzu im IBM Informix Dynamic Server Performance Guide:
Fragmenting Indexes
(siehe Weblinks 1.)

TechTipp: Functional Index

Sobald Funktionen auf Datenfelder angewandt werden, kann die Suche nicht mehr über den Index erfolgen. Ein typischer Fall hierfür ist der Vergleich von Namen mit UPPER(). Für diesen Fall kann ein Functional Index angelegt werden, der dann bei Abfragen genutzt wird. Zu beachten ist hierbei, dass ein solcher Index auf eine Function nur dann erstellt werden kann, wenn die Function mit (NOT VARIANT) erstellt wurde.
Beispiel:

```
CREATE FUNCTION UPSHIFT( in_str VARCHAR(255) )
     RETURNING VARCHAR(255)
     WITH( NOT VARIANT )
DEFINE out_str VARCHAR(255);
let OUT_STR=UPPER(in_str);
RETURN( out_str );
END FUNCTION:

CREATE INDEX I_CUST1 ON CUSTOMER( UPSHIFT( lname ));
UPDATE STATISTICS FOR TABLE CUSTOMER;

SELECT * FROM customer WHERE UPSHIFT( lname ) = "MAYER";
```

Mehr Infos hierzu im IBM Informix Guide to SQL: Syntax (siehe Weblinks 1.)
Using a Function as an Index Key (IDS)

TechTipp: DELIMIDENT

Identifier bezeichnen Datenbankobjekte wie Tabellen, Spalten oder Trigger.
Normalerweise gelten die folgenden Regeln, dann handelt es sich um einen „regular identifier“:

- ❑ Er darf nicht in einfache (') oder doppelte (") Anführungsstriche gesetzt werden
- ❑ Er darf nur mit einem Buchstaben oder einem Unterstrich beginnen
- ❑ Er darf nur Buchstaben, Zahlen, Unterstriche und, beim IDS auch das Dollarzeichen ($) erhalten.
  Allerdings wird die Verwendung des Dollarzeichens nicht empfohlen.
- ❑ Er darf kein reserviertes SQL Schlüsselwort sein

Regular Identifier werden automatisch in Kleinbuchstaben umgewandelt. Die regular Identifier MyTab, MYTAB und mytab werden alle als der gleiche Identifier mytab behandelt.
Werden im Identifier andere Zeichen benötigt, oder soll zwischen Groß- und Kleinschreibung unterschieden werden, so kann dass durch die Kombination aus Setzen der Umgebungsvariable DELIMIDENT und verwenden von doppelten Anführungsstrichen erreicht werden.
Die Umgebungsvariable DELIMIDENT definiert also nur, dass es delimited Identifier gibt,

also Identifier, die durch ein besonderes Zeichen, die doppelten Anführungsstriche, abgegrenzt sind. Die Identifier selbst müssen dann in diesen Zeichen eingeschlossen werden, um erkannt zu werden.
Um also die Funktionalität von DELIMIDENT nutzen zu können, muss die Umgebungsvariable definiert sein. Die Variable kann sowohl in der Umgebung des Servers als auch in der Umgebung des Clients gesetzt werden. Auf dem Server wird nur ausgewertet, ob DELIMIDENT gesetzt, in der Clientumgebung kann das Verhalten durch die Werte „Y“ und „N“ genauer bestimmt werden. Dieses Verhalten von DELIMIDENT in der Clientumgebung wird in einen eigenen Artikel beschrieben.

Delimited Identifier erlauben es, reservierte Worte als Namen von Datenbankobjekten zu verwenden. Sie können einen delimited Identifier nicht als Datenbank Namen verwenden. Er darf nur Zeichen enthalten, die Teil des Codeset für die gesetzte db_locale sind. Delimited Identifer unterscheiden zwischen Groß- und Kleinschreibung. Damit bezeichnen "MyTab" und "mytab" zwei unterschiedliche Tabellen. Allerdings ist es empfehlenswert, die Datenbankobjekte deutlicher als durch Groß- und Kleinschreibung voneinander zu unterscheiden.
Wenn innerhalb eines delimited Identifier doppelte Anführungszeichen verwenden werden sollen, müssen diese doppelt angeben werden (""). Diese Zeichenfolge wird dann als ein einziges doppeltes Anführungszeichen interpretiert.

Die folgende Tabelle zeigt, wie Leerzeichen und Anführungszeichen in Abhängigkeit von der Umgebungsvariable DELIMIDENT interpretiert werden.

Syntax	**DELIMIDENT nicht gesetzt**		**DELIMIDENT gesetzt**	
	Identifier	**Ergebnis**	**Identifier**	**Ergebnis**
`CREATE TABLE My Tab`	My = identifier Tab = unbekanntes Keyword	Syntax Error	My = identifier Tab = unbekanntes keyword	Syntax Error
`CREATE TABLE "My Tab"`	illegale Zeichen (" und Leerzeichen)	Syntax Error	My Tab	Tabelle My Tab wird angelegt
`CREATE TABLE "MyTab"`	Illegale Zeichen (")	Syntax Error	MyTab	Tabelle MyTab wird angelegt
`CREATE TABLE MyTab`	Mytab	Tabelle mytab wird angelegt	Regular Identifier mytab	Tabelle mytab wird angelegt
`CREATE TABLE "My""Tab"`	Illegale Zeichen (")	Syntax Error	My"Tab	Tabelle My"Tab wird angelegt

Tipp: WebLinks rund um INFORMIX:

1. OnlineHilfe (Schnelles Nachschlagen in den Handbüchern):
   http://publib.boulder.ibm.com/infocenter/idshelp/v10/index.jsp
2. Redbooks (Nützliche Anleitungen mit praktischen Tipps):
   http://www.redbooks.ibm.com/
3. Linux (Wichtige Hinweise zur Installation unter Linux):
   http://www-306.ibm.com/software/data/informix/linux/ids.html
4. Die deutsche INFORMIX User Group
   http://www.iug.de
5. International INFORMIX User Group
   http://www.iiug.org

Bericht zur IBM IOD Konferenz in Anaheim (USA)

Die erste globale IBM IOD Konferenz in Anaheim (USA): auch aus Informix Sicht ein voller Erfolg!

Mit mehr als 5000 Teilnehmern, davon über 1000 IBM Geschäftspartnern und über 60 Informix Sessions war die erste IBM Information On Demand Konferenz 2006 sehr erfolgreich.
Besonders aus deutscher Informix Sicht sind wir ganz besonders stolz, dass wir zwei Kunden- und einen Partner-Sprecher aus Deutschland für die Konferenz gewinnen konnten, die drei sehr interessante Vorträge präsentierten.
Darüberhinaus wurde der deutsche Informix OEM Distributor, die Bytec Bodry Technology GmbH als einziger IBM Geschäftspartner während der Konferenz mit zwei Auszeichnungen geehrt.
Sehr interessant: von fünf IBM IOD Auszeichnungen gingen vier an Informix Partner weltweit.
Sie können demnächst alle diese Details und noch mehr in einem interessanten Reisebericht von Alexander Koerner nachlesen.
Der Bericht wird in den nächsten Tagen unter „Aktuelles“ auf folgender Seite zu finden sein:
http://www-306.ibm.com/software/de/db2/informix/ids.html

Die Events der nächsten 6 Wochen auf einen Blick

Folgende Kurse zu INFORMIX finden in diesem Jahr noch statt:

T254DE System Monitoring Interface	04.12. - 08.12.	München
T291DE Datenbankadministration	11.12. - 15.12.	Düsseldorf
T280DE Einführung in SQL	06.12. - 08.12.	Stuttgart

Die Autoren dieser Ausgabe:

Dr. Elisabeth Bach	IT Specialist, Informix Advanced Support
Alexander Koerner	Certified Senior IT-Specialist
Gerd Kaluzinski	Consultant, Software Group, Information Management
Sandor Szabo	Manager IBM Informix Database Development
Thomas Simoner	OEM Sales Manager, Informix und DB2