



Firebird Database Statistics Reporting Tool

Norman Dunbar, Mark Rotteveel, Martin Köditz

Version 1.6-de, 31. Juli 2020

Inhaltsverzeichnis

1. Einleitung	2
2. Befehlszeilenooptionen	3
3. Gstat Examples And Interpretation	7
3.1. Datenbank-Header	7
3.2. Analysieren der gesamten Datenbank	12
3.3. Nur Datenseiten analysieren	12
3.4. Nur Indexseiten analysieren	14
3.5. Auswählen der zu analysierenden Tabellen	18
3.6. Systemtabellen und -indizes einschließen	19
3.7. Datensatz- & Versionsdetails	19
3.8. Wenn Sie eine beschädigte Datenbank haben	22
4. gstat-Fallstricke	23
4.1. Der Schalter -t[able] kann Probleme machen	23
4.2. Die Schattenzahl scheint falsch zu sein	24
Anhang A: Dokumentenhistorie	26
Anhang B: Lizenzhinweis	27

Kapitel 1. Einleitung

gstat ist eines der mit Firebird gelieferten Datenbankdienstprogramme. Es wird verwendet, um statistische Details zum Inhalt einer Datenbank anzuzeigen. gstat stellt keine Verbindung zur Datenbank her wie andere Dienstprogramme, sondern öffnet die Datenbankdatei(en) direkt und liest die Rohdaten. Aus diesem Grund ist gstat nicht transaktionsbewusst und einige der gesammelten Statistiken können Daten enthalten, die beispielsweise durch normale Datenbanktransaktionen gelöscht wurden.

In diesem Handbuch diskutieren wir:

- Befehlszeilenoptionen für gstat.
- gstat Befehle und ihre Parameter.
- Ausführen von gstat und interpretieren der Ergebnisse.
- Einige Vorbehalte, Fallstricke und Schwächen von gstat.

Kapitel 2. Befehlszeilenoptionen

gstat sollte entweder als Root oder als Firebird-Benutzer ausgeführt werden. Dies liegt daran, dass die Standardberechtigungen des Betriebssystems beim Erstellen einer neuen Datenbank so sind, dass nur der Eigentümer — firebird — Zugriff auf die Datenbankdatei(en) hat. Selbst Mitglieder der Firebird-Gruppe haben standardmäßig keinen Lesezugriff.

gstat wird normalerweise wie folgt aufgerufen:

```
gstat database_name [switches]
```

In einigen Dokumentationen wird empfohlen, gstat wie folgt aufzurufen:

```
gstat [switches] database_name
```

Obwohl dies auf diese Weise funktioniert, treten Probleme auf, wenn der Schalter `-t[able]` verwendet wird.

Der Datenbankname kann keine entfernte Datenbank sein, er muss lokal sein, er kann jedoch ein Alias für eine lokale Datenbank sein. Der Grund, warum es lokal sein muss, ist, dass gstat auf der Ebene der *physischen Datei* funktioniert, anstatt eine Datenbankverbindung zum Server herzustellen — es liest die Datenbankdatei direkt.

Wenn gstat mit einem ungültigen Schalter oder mit dem neuen Schalter `-?` ab Firebird 2.5 aufgerufen wird, wird Folgendes angezeigt, um Sie an die gültigen zu erinnern. Leider wird nur die Kurzform der Schalter angezeigt.

```
./gstat -?
usage:  gstat [options] <database> or gstat <database> [options]
Available switches:
  -a      analyze data and index pages
  -d      analyze data pages
  -h      analyze header page ONLY
  -i      analyze index leaf pages
  -s      analyze system relations in addition to user tables
  -u      username
  -p      password
  -fetch  fetch password from file
  -r      analyze average record and version length
  -t      tablename <tablename2...> (case sensitive)
  -z      display version number
option -t accepts several table names only if used after <database>
```



In Firebird-Versionen vor 2.0 kann auch der Schalter `-l[og]` verwendet werden. Dieser berichtet über die Details der Protokollierungsseite(n) in der Datenbank. Die Protokollierungsseiten werden seit einiger Zeit nicht mehr verwendet und der Schalter wurde aus `gstat` entfernt.



Der `-fetch`-Schalter ist erst ab Firebird 2.5 verfügbar.

Diese Schalter werden unten beschrieben.

`-a[ll]`

Dies ist der Standardschalter und entspricht `-h[earer] -d[ata] -i[index]`. In Abwesenheit von `-d[ata]` und `-i[index]` wird `gstat` so ausgeführt, als ob beide neben `-h[earer]` angegeben worden wären.

`-d[ata]`

Wenn Sie diesen Schalter angeben, analysiert `gstat` jede Benutzertabelle in der angegebenen Datenbank. Benutzerindizes, Systemtabellen und Systemindizes werden nicht analysiert.

`-h[earer]`

Dieser Schalter zeigt Statistiken über die Datenbank selbst an und wird dann beendet. Die Header-Informationen werden auch angezeigt, wenn ein anderer Schalter verwendet wird. So erhalten Sie in Ihrer Ausgabe immer Details zum Datenbank-Header.

`-i[ndex]`

Wenn Sie diesen Schalter angeben, analysiert `gstat` jeden Benutzerindex in der angegebenen Datenbank. Benutzertabellen, Systemindizes und Systemtabellen werden nicht analysiert.

`-s[ystem]`

Dieser Schalter ist ein Modifikator und ändert die Ausgabe der Schalter `-d[ata]` oder `-i[ndex]`, indem die Systemtabellen (oder Indizes) zusätzlich zu den benutzerdefinierten Tabellen (oder Indizes) eingeschlossen werden. Die alleinige Verwendung dieses Schalters entspricht dem Aufruf von `gstat` mit dem angegebenen `-a[ll] -s[ystem]`.

Bei der Ausführung listet dieser Schalter Statistiken für die verschiedenen RDB\$-Tabellen und -Indizes auf, und wenn er gegen Firebird 2 ausgeführt wird, auch für die verschiedenen MON\$-Tabellen und -indizes.

`-r[ecord]`

Der Schalter `-r[ecord]` ist ein Modifikator für die Schalter `-d[ata]` und `-s[ystem]`. Es werden Daten zu den durchschnittlichen Datensatz- und Versionslängen für alle analysierten Datentabellen (Benutzer und / oder System) hinzugefügt. Dieser Schalter hat keine Auswirkung auf den Schalter `-i[ndex]`.

`-t[able]`

Mit diesem Schalter können Sie eine Tabelle oder eine Liste von Tabellen sowie alle zu den angegebenen Tabellen gehörenden Indizes analysieren. Im Abschnitt `<< gstat-caveats,Fallstricke>>` unten finden Sie einige mögliche Probleme mit diesem Schalter und ein Beispiel für dessen Verwendung.

Dem Schalter `-t[able]` sollte eine Liste der Tabellennamen folgen, die Sie analysieren möchten. Die Liste muss in Großbuchstaben geschrieben sein und jede Tabelle ist durch ein Leerzeichen getrennt. Es ist auch möglich, doppelte Anführungszeichen zu verwenden, um `gstat` zu veranlassen, eine Tabelle zu analysieren, deren Name nicht in Großbuchstaben geschrieben ist.

Es ist nicht erforderlich, den Schalter `-i[ndex]` anzugeben, da alle Indizes in den angegebenen Tabellen analysiert werden. Die Datenbankheaderinformationen werden ebenfalls angezeigt.

`-u[sername]`

Ermöglicht die Angabe des Benutzernamens des SYSDBA- oder Datenbankeigentümers. Dies muss nicht angegeben werden, wenn die Umgebungsvariable `ISC_USER` vorhanden ist und einen korrekten Wert für den Benutzernamen hat oder wenn Sie als privilegiertes Konto am Server angemeldet sind.



Ein privilegiertes Konto ist eines der folgenden:

- Wurzel
- Feuervogel
- Interbase
- interbas (ohne das letzte 'e')

Wenn Sie sich mit einem dieser Konten beim Server anmelden, erhalten Sie automatisch SYSDBA-Berechtigungen. Wenn Sie ein anderes Konto verwenden, müssen Sie möglicherweise einen Benutzernamen und ein Kennwort angeben, um `gstat` auszuführen.

`-p[assword] <password>`

Liefert das Passwort für den oben angegebenen Benutzernamen. Dies muss nicht angegeben werden, wenn die Umgebungsvariable `ISC_PASSWORD` vorhanden ist und den richtigen Wert hat oder wenn Sie mit einem privilegierten Konto am Server angemeldet sind.

`-fetch <password file name> | stdin | /dev/tty`

Dieser Schalter bewirkt, dass das Kennwort für den entsprechenden Benutzer aus einer Datei gelesen wird und nicht in der Befehlszeile angegeben wird. Der angegebene Dateiname ist *not* in Anführungszeichen und muss für den Benutzer lesbar sein, der `gstat` ausführt. Wenn der Dateiname als `stdin` angegeben ist, wird der Benutzer zur Eingabe eines Kennworts aufgefordert. Auf POSIX-Systemen führt der Dateiname `/dev/tty` auch zu einer Aufforderung zur Eingabe des Kennworts.

HINWEIS: Ab Firebird 2.5.

`-z`

Dies ist ein Modifikatorschalter. Mit `-z` wird die Versionsnummer des Dienstprogramms `gstat` und der Firebird-Installation angezeigt. Sie müssen einen gültigen Datenbanknamen und möglicherweise einen weiteren Schalter angeben. Dieser Schalter fügt die Details der Version `gstat` und Firebird zum Ausgang des anderen von Ihnen bereitgestellten Schalters hinzu — oder die Standardeinstellung, wenn Sie keinen angegeben haben. Die kürzeste Ausgabe würde von

einem `-t non_existent_tablename` stammen, wenn Sie nur die Versionsdetails wie folgt benötigen:

```
tux> gstat -t non_existing_tablename -z employee
gstat version LI-V2.1.3.18185 Firebird 2.1

Database "/opt/firebird/examples/empbuild/employee.fdb"
Database header page information:
...

Database file sequence:
File /opt/firebird/examples/empbuild/employee.fdb is the only file
    Firebird/linux Intel (access method), version
"LI-V2.1.3.18185 Firebird 2.1"
    Firebird/linux Intel (remote server), version
"LI-V2.1.3.18185 Firebird 2.1/tcp (greenbird)/P11"
    Firebird/linux Intel (remote interface), version
"LI-V2.1.3.18185 Firebird 2.1/tcp (greenbird)/P11"
    on disk structure version 11.1

Analyzing database pages ...
```

HINWEIS: Die obige Ausgabe wurde geringfügig geändert, damit sie der Seitenbreite eines PDF-Dokuments entspricht.

Die Ausgabe beginnt mit der Anzeige der `gstat`-Version, gefolgt von den Details des Datenbank-Headers. Als nächstes werden die Datenbankdatei und die Firebird-Details angezeigt und schließlich die Details für den angegebenen Tabellennamen, die natürlich nicht gefunden werden.

Kapitel 3. Gstat Examples And Interpretation

Dieser Abschnitt enthält häufig ausgeführte Statistiksammlungen und erläutert die Ausgabe.

3.1. Datenbank-Header

Diese Option erzeugt die geringste Ausgabemenge — es sei denn, Sie geben einen einzelnen nicht vorhandenen Tabellennamen mit dem Schalter `-t[table]` an — und ist in allen anderen Schaltern enthalten, sodass dies zuerst erläutert wird.

```
tux> gstat employee -header

Database "/opt/firebird/examples/empbuild/employee.fdb"
Database header page information:
  Flags                0
  Checksum             12345
  Generation           184
  Page size            4096
  ODS version          11.1
  Oldest transaction   166
  Oldest active        167
  Oldest snapshot      167
  Next transaction     170
  Bumped transaction   1
  Sequence number      0
  Next attachment ID   68
  Implementation ID    19
  Shadow count         0
  Page buffers         0
  Next header page     0
  Database dialect     3
  Creation date        Sep 25, 2009 12:50:24
  Attributes           multi-user maintenance

Variable header data:
  Sweep interval:     20000
  *END*
```

In der ersten Ausgabezeile werden die Dateinamen und der Pfad der Datenbank angezeigt. Dies kann nützlich sein, um einen Datenbankalias aufzulösen und herauszufinden, wo sich die Datenbank befindet. Da es sich bei der Mitarbeiterdatenbank um eine Einzeldateidatenbank handelt, wird nur eine Datei angezeigt. Wäre dies eine Datenbank mit mehreren Dateien gewesen, würde das Ende der obigen Auflistung wie folgt aussehen:


```
...  
Variable header data:  
Continuation file:      /u00/firebird/databases/multi_employee.fdb1  
Last logical page:     162
```

Die Details der verschiedenen Headerfelder werden unten beschrieben:

Flags

Flags werden auf einer Datenbank-Headerseite nicht verwendet.

Checksum

Alle Prüfsummen sind 12345. Prüfsummen auf den verschiedenen Datenbankseiten werden nicht mehr verwendet.

Generation

Die Generierungsnummer wird jedes Mal erhöht, wenn diese Seite in die Datenbank neu geschrieben wird.

Page size

Die Seitengröße der gesamten Datenbank. Da die Datenbankdatei in verschiedene Seiten aufgeteilt werden muss, kann der SYSDBA bei der Erstellung angeben, wie groß die gewünschte Seitengröße sein soll. Jede Seite in der Datenbank hat dieselbe Größe.

ODS version

Die On-Disc-Struktur einer Datenbank definiert möglicherweise zusammen mit dem SQL-Dialekt, welche Funktionen des Firebird-Datenbanksystems Benutzern dieser Datenbank zur Verfügung stehen. Diese Funktionen sind möglicherweise in der von Ihnen ausgeführten Version von Firebird vorhanden. Wenn das Datenbank-ODS jedoch älter ist, sind einige der neuen Funktionen nicht verfügbar.

Werte, die Sie derzeit hier sehen können, sind:

- 5.0 für Interbase 3.3
- 8.0 für Interbase 4.0
- 9.0 für Interbase 4.5
- 9.1 für Interbase 5.0
- 10.0 für Firebird 1.0 und Interbase 6.0
- 10.1 für Firebird 1.5
- 11.0 für Firebird 2.0
- 11.1 für Firebird 2.1
- 11.2 für Firebird 2.5

Transaction details

Der Bericht enthält eine Reihe verschiedener Transaktionsdetails. diese sind:

Oldest transaction

Die Transaktions-ID der sogenannten *Oldest Interesting Transaction* oder OIT. Dies ist einfach die ID der am längsten laufenden Transaktion, die bisher nicht über ein *hartes* Commit abgeschlossen wurde. Es wurde möglicherweise zurückgesetzt (Rollback) oder befindet sich in der Schwebe (Limbo), aber wenn es per Commit festgeschrieben wurde, ist es nicht mehr interessant. Dieser Wert wird zusammen mit der ältesten Snapshot-Transaktion verwendet, um festzustellen, ob ein automatischer Sweep der Datenbank erforderlich ist.



Es gibt zwei Commits — Commit und Commit-Retaining (Beibehaltung). Nur das erste davon ist ein harter Commit, das bei Ausführung die Transaktion als nicht mehr interessant macht. Durch die Beibehaltung des Commits bleibt die Transaktion weiterhin interessant. Einige Datenbankdienstprogramme und/oder -tools, die ein Commit durchführen, führen tatsächlich ein Commit-Retaining durch, wodurch Ihre Datenbank mit vielen noch interessanten Transaktionen belassen werden kann.

Oldest active

Die ID der ältesten *aktiven* Transaktion oder OAT. Dieser Wert zeigt die Transaktions-ID (TID) der ältesten noch laufenden Transaktion an. Eine Transaktion gilt als aktiv, wenn sie nicht durch einen *harten* Commit festgeschrieben wurde, sich nicht in einem Schwebezustand befindet und nicht zurückgesetzt wurde.

Oldest snapshot

Die ID der ältesten Transaktion, die derzeit nicht zur Müllabfuhr (Garbage Collection) berechtigt ist. Bei Transaktionen mit dieser oder einer höheren ID können beispielsweise noch keine alten Datensatzversionen durch einen Sweep entfernt werden. Normalerweise entspricht dies dem obigen OAT. Die Differenz zwischen *diesem* Wert und der OIT, wenn sie größer als das Datenbank-Sweep-Intervall ist — vorausgesetzt, das automatische Sweeping ist nicht deaktiviert — bestimmt, ob ein automatischer Sweep stattfindet.



Viele Websites, Bücher und Handbücher (zuvor einschließlich dieses) erklären, dass der automatische Sweep aktiviert wird, wenn $OAT - OIT$ größer als das Sweep-Intervall ist. Vlad Khorsun, einer der Firebird-Entwickler, erklärte, dies ist *nicht* der Fall, wenn der Sweep aktiviert wird, wenn $OST - OIT$ größer ist als der Schwellenwert.

Das heißt: Der automatische Sweep wird gestartet, wenn die Differenz zwischen OST (Oldest Snapshot Transaction) und OIT größer als das definierte Sweep-Intervall ist. Der Prozess des Benutzers, der versucht hat, die Transaktion zu starten, die das Sweep-Intervall um eins überschreitet, durchsucht die gesamte Datenbank, bevor die angeforderte Transaktion tatsächlich gestartet wird.

Next transaction

Die nächste in der Datenbank gestartete Transaktion hat diese ID-Nummer.

Bumped transaction

Immer 1, nicht mehr verwendet.

Wenn Sie feststellen, dass der Unterschied zwischen der OAT und der ID der nächsten Transaktion immer größer zu werden scheint, wird etwas in Ihrer Datenbank nicht ordnungsgemäß festgeschrieben, sodass sich möglicherweise eine zunehmende Anzahl von Speicherbereinigungen aufbaut. Schließlich werden Sie feststellen, dass die Datenbankstartzeiten immer länger dauern und die Leistung immer langsamer wird. Überprüfen Sie die Zahlen, und wenn ein Problem festgestellt wird, sollten Sie `gfix` ausführen, um manuell einen Datenbank-Sweep auszuführen, um den Müll zu beseitigen und die normale Arbeit in der Datenbank wiederherzustellen.

Weitere Informationen zum Erkennen und Behandeln von Transaktionen in der Schwebe finden Sie im Abschnitt *Limbo Transaction Management* im Handbuch `gfix`. Dies kann sich durchaus auf die Fähigkeit des Datenbank-Sweep-Prozesses auswirken, alte redundante Daten aus älteren uninteressanten Transaktionen zu entfernen. Limbo-Transaktionen werden verursacht, wenn ein zweiphasiges Commit über mehrere Datenbanken aus irgendeinem Grund fehlschlägt. Limbo-Transaktionen sind für die Datenbank immer noch interessant und müssen mit `gfix` festgeschrieben oder zurückgesetzt werden, da die Sweep-Verarbeitung nicht erkennen kann, ob dies ohne menschliches Eingreifen sicher ist oder nicht.

Sequence number

Immer Null. Dies war die Sequenznummer der Datenbank-Headerseite, wird jedoch nicht mehr verwendet.

Next attachment ID

Die ID-Nummer des nächsten Anhangs zu dieser Datenbank. Jedes Mal, wenn eine Anwendung eine Verbindung zur Datenbank herstellt, steigt diese Zahl um eins. Durch das Starten und Herunterfahren der Datenbank wird auch diese Anzahl erhöht. `gstat`-Verbindungen ändern die ID nicht, da sie sich nicht auf normale Weise verbinden.

Implementation ID

Wenn die Datenbank erstellt wurde, wurde sie möglicherweise auf einem anderen System erstellt — Hardware, Betriebssystem usw. — als dem, auf dem sie jetzt ausgeführt wird. Die Implementierungs-ID zeigt Ihnen, auf welcher Hardwarearchitektur die Datenbank ursprünglich erstellt wurde.

Die Implementierungs-ID wird verwendet, um zu bestimmen, ob die Datenbank tatsächlich auf der Hardware verwendet werden kann, auf der sie gerade ausgeführt wird, oder ob es eine Funktion der ursprünglichen Hardware gibt, auf der die Datenbank erstellt wurde, die sie mit dem aktuellen Hostsystem inkompatibel macht.

Shadow count

Zeigt die Anzahl der an diese Datenbank angehängten oder für diese Datenbank verfügbaren Schattendateien an. Manchmal ist dieser Wert falsch, selbst wenn kürzlich Schattendateien erstellt und/oder gelöscht wurden.



Aufgrund der Inkonsistenz zwischen den Berichten von gstat und der Realität ist es am besten, isql und den Befehl SHOW DATABASE zu verwenden, um die korrekten Details der Schattendateien anzuzeigen.

Page buffers

Wenn dieser Wert als Null angezeigt wird, verwendet die Datenbank den Standardwert des Servers für die Anzahl der Seiten, die während des Betriebs der Datenbank im Speicher zwischengespeichert werden können. Die Einstellung kann in der Datei firebird.conf definiert werden. Unter Firebird Superserver 2.1 ist diese Einstellung DefaultDbCachePages in der Konfigurationsdatei und auf 2048 Seiten festgelegt. Sie können gfix verwenden, um dies zu ändern, ohne die Konfigurationsdatei zu bearbeiten.

Database dialect

Die SQL-Dialektnummer der Datenbank. Normalerweise 1 oder 3. Diese Einstellung kann mit gfix geändert werden und hilft neben dem ODS-Wert zu bestimmen, welche Funktionen von Firebird verfügbar sind, wenn Anwendungen die Datenbank verwenden.

Creation date

Das Datum, an dem diese Datenbank ursprünglich erstellt wurde. Es kann das Datum anzeigen, an dem die Datenbank zuletzt von gbak wiederhergestellt wurde.

Attributes

In diesem Teil des Berichts werden Informationen zu verschiedenen Attributen der Datenbank angezeigt. Beispiele für das, was Sie möglicherweise sehen, sind:

no reserve

Alle Seiten werden zu 100% gefüllt und sind in schreibgeschützten Datenbanken am nützlichsten. Auf jeder Seite ist kein Platz für Aktualisierungen und / oder Löschungen reserviert.

force write

Schreibvorgänge auf Festplatten werden nicht zwischengespeichert. Sie werden zum Zeitpunkt der Schreibanforderung auf die Hardware geschrieben. Dies wird hauptsächlich in Windows-Datenbanken verwendet, in denen das Cache-Verwaltungssystem zu Schreibverlust und Datenbankbeschädigung führen kann.

shutdown

Die Datenbank wurde geschlossen und kann nicht verwendet werden.

read only

Die Datenbank wird im schreibgeschützten Modus ausgeführt.

multi-user maintenance

Die Datenbank ist wegen Wartungsarbeiten geschlossen. Mehrere Verbindungen sind nur von SYSDBA oder dem Datenbankeigentümer zulässig.

single-user maintenance

Die Datenbank ist wegen Wartungsarbeiten geschlossen. Es ist nur eine SYSDBA- oder Datenbankigentümergebenisverbindung zulässig.

Abhängig von der verwendeten Version von Firebird und natürlich zukünftigen Versionen können hier andere Werte angezeigt werden.

Variable header data

Dieser Teil des Berichts behandelt Informationen, die sich nicht im festen Teil des Datenbankheaders befinden. Beispielsweise wird hier das Sweep-Intervall angezeigt und es werden Informationen zu etwaigen angehängten Sekundärdateien angezeigt. Wenn Sie die Datenbank beispielsweise mit dem Tool nbackup gesichert haben, werden hier Details der Sicherungs-GUID angezeigt — jedoch nur für die letzte Sicherung.

3.2. Analysieren der gesamten Datenbank

Die Analyse der gesamten Datenbank ist die Standardeinstellung für gstat. Bei Verwendung werden alle Benutzertabellen und -indizes analysiert und die gesammelten Statistiken gemeldet. Da die Ausgabe höchstwahrscheinlich sehr groß sein wird, ist es ratsam, die Ausgabe in eine Datei zu leiten:

```
gstat employee >employee.gst
```

Die Ausgabe besteht aus einer Analyse jeder Benutzertabelle und aller zugehörigen Benutzerindizes. Die Interpretation dieser Ergebnisse wird nachstehend in den Abschnitten zur Analyse von Daten und Indexseiten behandelt.

3.3. Nur Datenseiten analysieren

Der Befehl, nur Benutzertabellen in der Datenbank zu analysieren, lautet:

```
gstat employee -data >employee.gst
```

Die mit diesem Befehl ausgegebenen Ergebnisse listen die Benutzertabellen in alphabetischer Reihenfolge auf. Es werden keine Indizes analysiert oder aufgelistet, unabhängig davon, wie viele in der Datenbank vorhanden sind.

Sobald der Bericht fertiggestellt ist, können die Ergebnisse wie folgt analysiert werden, wobei insbesondere eine Tabelle betrachtet wird.

CONFIGREVISIONSTORE (213)

Primary pointer page: 572, Index root page: 573

Data pages: 2122, data page slots: 2122, average fill: 82%

Fill distribution:

0 - 19% = 1

20 - 39% = 0

40 - 59% = 0

60 - 79% = 79

80 - 99% = 2042

Der obige Auszug aus dem Bericht beginnt mit der Anzeige des Tabellennamens—CONFIGREVISIONSTORE—und der Tabellen-ID—213. Die ID der Tabelle ist tatsächlich die Spalte RDB\$RELATION_ID in der Systemtabelle RDB\$RELATIONS, wie die folgende isql-Sitzung zeigt:

```
SQL> select rdb$relation_name
CON> from rdb$relations
CON> where rdb$relation_id = 213;
```

```
RDB$RELATION_NAME
=====
CONFIGREVISIONSTORE
```

Primary pointer page

Dies ist die Seitenzahl innerhalb der Datenbank der ersten Seite mit Zeigern auf die Datenseiten dieser Tabelle. Die Struktur der Datenbank ist so, dass jede Tabelle exklusive Datenseiten enthält und eine Liste dieser Seiten irgendwo aufbewahrt werden muss. Diese Statistik gibt Ihnen die Seitenzahl für diesen Ort.

Index root page

Dies ist die Seitenzahl, auf der sich die erste Seite mit Zeigern auf die Tabellenindizes in der Datenbank befindet. Jede Tabelle in der Datenbank hat eine Seite, die Indexstammseite, die Zeiger auf die Apex-Seiten für jeden einzelnen Index enthält.

Data pages

Die Gesamtzahl der dieser Tabelle zugewiesenen Seiten. Da gstat keine transaktionsbezogene Verbindung zur Datenbank herstellt, kann es nicht feststellen, ob es sich bei einer dieser Seiten um alte Datensatzversionen (Garbage) oder um gelöschte Datensätze in derzeit nicht festgeschriebenen Transaktionen handelt. Daher ist die Anzahl möglicherweise höher als erforderlich da diese zusätzlichen Seiten in der Gesamtsumme enthalten sind.

Data page slots

Dieser Wert sollte der Anzahl der Datenseiten entsprechen. Es gibt die Anzahl der Zeiger auf Seiten in dieser Tabelle an, die auf verschiedenen Zeigerseiten innerhalb der Datenbank gespeichert sind. Wenn sich die Zahlen unterscheiden, kann dies an dem Müll liegen, der nicht gesammelt wird.

Average fill

Der berechnete Speicherplatz, der durchschnittlich auf jeder Seite der Tabelle verwendet wird. Die Abbildung enthält Speicherplatz, der von früheren Versionen von Datensätzen in der Tabelle verwendet wird. Die Füllverteilung (unten) enthält weitere Details.

Fill distribution

In diesem Abschnitt des Berichts wird ein 5-Band-Histogramm angezeigt, in dem jedes Band 20% des auf jeder Seite ausgefüllten Speicherplatzes darstellt. Im obigen Beispiel sehen wir, dass diese Tabelle eine einzelne Seite enthält, die zu weniger als 20% gefüllt ist. 79 Seiten sind zu 60% bis 79% gefüllt, während die überwiegende Mehrheit, 2042, zu 80% bis 99% gefüllt ist.

3.4. Nur Indexseiten analysieren

Der Befehl, nur Benutzerindizes in der Datenbank zu analysieren, lautet:

```
gstat employee -index >employee.gst
```

Die mit diesem Befehl ausgegebenen Ergebnisse listen die Benutzertabellen in alphabetischer Reihenfolge auf. Es werden keine Tabellen analysiert. Der Bericht listet jedoch die Tabellennamen in alphabetischer Reihenfolge auf und listet alle anwendbaren Indizes unter dem entsprechenden Tabellennamen auf.

Nach Abschluss der Analyse können die Ergebnisse wie folgt interpretiert werden. Das folgende Beispiel zeigt die Ausgabe eines einzelnen Index in einer Datenbank.

```
CONFIGREVISIONSTORE (213)
  Index PK_CONFIGREVISIONSTORE (0)
    Depth: 3, leaf buckets: 174, nodes: 62372
    Average data length: 2.58, total dup: 0, max dup: 0
    Fill distribution:
      0 - 19% = 15
      20 - 39% = 0
      40 - 59% = 55
      60 - 79% = 68
      80 - 99% = 36
```

Der obige Auszug aus dem Bericht beginnt mit der Anzeige des Tabellennamens — CONFIGREVISIONSTORE — und der Tabellen-ID — 213 wie oben beschrieben.

Nach den Details der Tabelle — und nur der Name und die ID werden angezeigt — werden die Indexdetails angezeigt. Wie oben werden der Indexname und seine ID angezeigt. Dieses Mal bezieht sich die ID auf die Position des Index in der Liste aller in der Tabelle erstellten Indizes. ID Null ist der erste erstellte Index, ID 1 ist der nächste und so weiter. Die Ausgabe von gstat listet die Indizes möglicherweise nicht in der ID-Reihenfolge auf. Wenn Indizes erstellt, aber anschließend gelöscht wurden, kann es zu Lücken in der ID-Sequenz kommen.

In den nächsten beiden Zeilen nach dem Indexnamen und der ID werden die Gesamtstatistiken für

diesen Index angezeigt.

Depth

Diese Statistik zeigt die Anzahl der Seiten an, auf die zugegriffen werden muss, um zu einem Indexeintrag zu gelangen. In diesem Beispiel müssen wir drei separate Seiten in den Puffercache lesen, bevor wir die Indexdetails verwenden können, um auf die gewünschte Zeile in der Tabelle zuzugreifen. Dies wird häufig als Index-Indirektion bezeichnet.

```
Depth: 3
```

Auf der Festplatte befindet sich eine *Index-Stammseite* (Root Page) der obersten Ebene, die gleichzeitig mit der Datenbank erstellt wird. Diese Seite enthält eine Liste von Zeigern auf die obere Seite (Apex) für jeden Index — eine Seite pro Index. Für jeden Index enthält diese Seite eine Liste mit Zeigern auf Folgendes:

- Apex-Seiten einer anderen Ebene, wenn die Tiefe größer als 1 ist, oder,
- zu den Blattseiten (Leaf Pages) für die tatsächlichen Indexdaten, wenn Tiefe = 1.

Auf den Blattseiten wird der Speicherort der indizierten Daten gespeichert. Die Indextiefe ist die Anzahl der Ebenen, die Sie von der Apex-Seite des Index herabsteigen müssen, um zu den Blattseiten zu gelangen. Weder die Indexstammseite noch die Apex-Seite des Index werden in der Tiefe gezählt.

Im Durchschnitt zeigt eine Tiefe von 2 oder weniger einen Index an, der effizient ist. Wenn die Tiefe 3 oder mehr beträgt, funktioniert der Index höchstwahrscheinlich nicht optimal. Die Lösung in dieser Situation besteht darin, `gbak` zu verwenden, um die Größe der Datenbankseite zu erhöhen, indem eine Sicherung erstellt und wie folgt wiederhergestellt wird:

```
tux> # Datenbank herunterfahren
tux> gfix -shut -tran 60 employee

tux> # Datenbank sichern
tux> gbak -backup employee /backups/employee.fbk

tux> # Aktuelle Seitengröße ermitteln
tux> gstat employee -header | grep -i "page size"
      page size          4096

tux> # Datenbank mit einer größeren Seitengröße wiederherstellen
tux> gbak -replace overwrite -page 8192 /backups/employee.fbk employee

tux> # Neue Seitengröße prüfen
tux> gstat employee -header | grep -i "page size"
      page size          8192

tux> # Datenbank hochfahren
tux> gfix -online normal employee
```


Sobald dies ausgeführt wurde, sollten Sie feststellen, dass die Tiefe des Index 2 oder weniger beträgt. Ist dies nicht der Fall, wiederholen Sie einfach den obigen Vorgang mit einer noch größeren Seitengröße.



Der obige Befehl zum Wiederherstellen der Sicherung *überschreibt* die ursprüngliche Datenbankdatei. Dies funktioniert, indem die Originaldatei gelöscht und neu erstellt wird. Sie müssen also wirklich sicherstellen, dass Ihre Datenbanksicherung tatsächlich funktioniert und dass die erstellte Sicherungsdatei verwendet werden kann, bevor Sie versuchen, eine Datenbank zu überschreiben. Weitere Informationen finden Sie im gbak-Handbuch.

Leaf buckets

Diese Statistik informiert uns über die Anzahl der Blattseiten, die dieser bestimmte Index verwendet. Eine Seite und ein Bucket sind synonym, aber Seite ist der modernere Begriff, der häufig verwendet wird.

```
leaf buckets: 174
```

In unserem Beispielindex sehen wir, dass die Datenbank 174 Seiten enthält, die die Details der indizierten Werte für diese Tabelle enthalten. Alle diese Seiten enthalten Zeiger auf die Daten.

Die Anzahl der Blattseiten sollte mit der Summe der Gesamtzahl der Seiten in jeder Histogrammleiste in der unten gezeigten Füllverteilung übereinstimmen.

Nodes

Dies ist die Gesamtzahl der Datensätze in der Tabelle, die indiziert wurden. Es ist jedoch möglich — da gstat nicht transaktionsbewusst funktioniert —, dass diese Zahl möglicherweise Zeilen enthält, die gelöscht (und nicht durch Müll gesammelt) wurden und / oder Datensätze mehr als zählen einmal, wenn sie so geändert wurden, dass die indizierten Spalten geändert wurden.

```
nodes: 62372
```

Aus den oben genannten Gründen ist es ratsam, vor dem Ausführen von gstat einen Sweep oder eine Datenbanksicherung und -wiederherstellung durchzuführen, um sicherzustellen, dass die gesammelten Statistiken korrekt sind und die wahre Position der Datenbank widerspiegeln.

Average data length

Diese Statistik gibt die durchschnittliche Länge der Schlüsselspalte(n) in Bytes an.

```
Average data length: 2.58
```

+ Dies ist höchstwahrscheinlich weniger als die tatsächliche Summe der Spaltengrößen, da Firebird die Indexkomprimierung verwendet, um die Datenmenge auf einer Indexblattseite zu reduzieren.

Duplicates

Duplikate sind in einem Primärschlüssel oder einem eindeutigen Index nicht zulässig. Andere Indizes erlauben Duplikate, und diese Statistiken geben Auskunft über die Anzahl der Duplikate, die der Index enthält. Die folgende isql-Abfrage zeigt die Details von Duplikaten für eine indizierte Spalte in einer anderen Tabelle als der bisher verwendeten — die keine Duplikate enthält.

```
SQL> SELECT IDX, COUNT(*)
CON> FROM NORMAN_TEST
CON> GROUP BY IDX;
```

IDX	COUNT
1	10
2	4
3	1

Von oben sehen wir insgesamt 15 Zeilen, von denen es 14 doppelte Werte gibt (alle mit einer 1 oder 2 in der IDX-Spalte). Das Folgende ist der Auszug für die Duplikate für diese Tabelle:

```
Index NORMANX (0)
  Depth: 1, leaf buckets: 1, nodes: 15
  Average data length: 0.27, total dup: 12, max dup: 9
```

Total dup ist die Gesamtzahl der Duplikate im Index. Beachten Sie, dass nur 12 Duplikate aufgelistet sind, wir jedoch bereits wissen, dass der Index 14 Duplikatzeilen enthält. Wie ist das möglich?

Das erste Auftreten einer 1 und das erste Auftreten einer 2 werden von gstat nicht als Duplikate gezählt. Nur die zweite und die nachfolgenden Kopien gelten als Duplikate.



Meiner Meinung nach ist dies kein ganz korrektes Verhalten. In der obigen Tabelle gibt es 15 Zeilen und nur drei eindeutige Werte in der IDX-Spalte, die indiziert ist. Mein Index enthält daher 14 doppelte Werte und nicht nur 12.

Sie können jedoch den Gesamt-Dup-Wert verwenden, um die Anzahl der eindeutigen Werte im Index zu extrahieren, indem Sie sie vom Knotenwert subtrahieren.

Max dup gibt die Anzahl der Indexeinträge an, die die längste Kette von Duplikaten gemeinsam haben. Mit anderen Worten — für den obigen Index — gibt es 9 Indexeinträge, die den gleichen Wert in der indizierten Spalte haben. Wir können sehen, dass dies wahr ist, da die Zeilen, in denen IDX 1 ist, 9 doppelte Einträge haben.

Wenn sich max dup dem Gesamtdup nähert, ist es eine vernünftige Annahme, dass der Index möglicherweise eine so geringe Selektivität aufweist, dass er möglicherweise nie in Abfragen verwendet wird.

Fill distribution

Der Rest des Berichts für unseren ursprünglichen Beispielindex zeigt, wie die Seiten im Index verwendet werden.

Fill distribution:

0	- 19%	= 15
20	- 39%	= 0
40	- 59%	= 55
60	- 79%	= 68
80	- 99%	= 36

Die Abbildungen stellen eine Grafik (oder ein Histogramm) dar, wie der Platz auf den Indexseiten genutzt wird. Jeder Wert des Histogramms repräsentiert die Anzahl der Seiten im gesamten Index, die bis zu einem bestimmten Prozentsatz gefüllt wurden. Jeder Balken des Histogramms repräsentiert den Prozentsatz, der für die Seite gefüllt ist.

Die Füllverteilung des Beispielindex ist oben dargestellt. Aus diesen Zahlen geht hervor, dass die überwiegende Mehrheit der Seiten zu 40 bis 99% gefüllt ist. Die einzelnen Zahlen am Ende jeder Zeile oben geben die Anzahl der Seiten in diesem Band an. Das Beispiel zeigt Folgendes:

- 15 pages have been filled to less than 20%; and
- 0 pages have been filled to between 20% and 39%; and
- 55 pages have been filled to between 40% and 59%; and
- 68 pages have been filled to between 60% and 79%; and
- 36 pages are filled to between 80% and 99%.

Die Summe aller dieser Seiten sollte sich zu der oben für Blattknoten gezeigten Zahl addieren.

Dieser Index zeigt eine recht gute Speicherplatznutzung, da die meisten Seiten gut gefüllt sind. Idealerweise möchten Sie, dass alle Seiten zu 80 bis 99% gefüllt sind. Wenn der Bericht andererseits zeigt, dass alle Seiten leicht gefüllt sind - beispielsweise weniger als 60% -, wäre der Index ein guter Kandidat für eine Wiederherstellungsübung.

Berücksichtigen Sie unbedingt die Gesamtzahl der Knoten, bevor Sie mit der Neuerstellung beginnen. Wenn der Index nur eine geringe Anzahl von Knoten enthält, hilft die Neuerstellung nicht bei der Speicherplatznutzung, da möglicherweise nicht genügend Datensätze vorhanden sind, um die Indexseiten tatsächlich zu füllen.

3.5. Auswählen der zu analysierenden Tabellen

Wenn Sie anstelle aller Benutzertabellen eine bestimmte Liste von Tabellen in die Analyse aufnehmen möchten, können Sie mit dem Schalter `-table` die Tabellen angeben, die Sie einbeziehen möchten. Beachten Sie, dass durch die Angabe von Tabellennamen auf diese Weise auch alle diesen Tabellen zugeordneten Indizes analysiert werden.

```
gstat employee -t EMPLOYEE JOB COUNTRY >employee.gst
```

Die resultierende Ausgabe wird wie oben beschrieben interpretiert.

Wenn Sie einen Tabellennamen haben, der von einem Benutzer erstellt wurde, der die Groß- und Kleinschreibung des Tabellennamens beibehalten möchte, anstatt ihn in Großbuchstaben konvertieren zu lassen, zum Beispiel:

```
tux> isql myMusic
Database: mymusic

SQL> CREATE TABLE "MyMusic_Artists" (
CON> art_id integer,
CON> art_name ....);

SQL> COMMIT;
```

... Dann müssen Sie die Tabellennamen in doppelten Anführungszeichen und in *genau* der gleichen Groß- und Kleinschreibung wie der Name der Tabelle in der Datenbank angeben:

```
gstat mymusic -t "MyMusic_Titles" "MyMusic_Artists" > MyMusic.gst
```

Wenn Sie einen nicht vorhandenen Tabellennamen angeben oder den Namen im falschen Fall usw. erhalten, ignoriert gstat ihn einfach.

3.6. Systemtabellen und -indizes einschließen

Bei der normalen Verwendung von gstat sind die Systemtabellen und -indizes nicht in der Ausgabe enthalten. Wenn Sie gstat mit dem Schalter `-system` aufrufen, werden diese Tabellen in die Analyse einbezogen.

```
gstat employee -system >employee.gst
```

Die Interpretation der Ergebnisse für die verschiedenen Systemtabellen und -indizes ist genau wie oben für Benutzertabellen und -indizes beschrieben.

3.7. Datensatz- & Versionsdetails

Wenn Sie gstat entweder mit den Standardschaltern oder `-d[ata]` oder `-t[able]` ausführen und den Schalter `-r[record]` hinzufügen, erhalten Sie zusätzliche Informationen im Bericht, in dem das angezeigt wird durchschnittliche Datensatzlänge und durchschnittliche Versionsdetails für die betreffende(n) Tabelle(n):

```
Average record length: 96.55, total records: 62372
Average version length: 0.00, total versions: 0, max versions: 0
```

Average record length

Einfach die durchschnittliche Datensatzlänge aller Datensätze in der Tabelle in Byte. Wenn diese Zahl 0,00 beträgt, können Sie ziemlich sicher sein, dass alle Ihre Datensätze gelöscht wurden oder dass Sie keine Datensätze in der Tabelle haben.

Total records

Die Gesamtzahl der Datensätze in der Tabelle. Der Wert kann Datensätze in aktuell aktiven Transaktionen enthalten und Datensätze enthalten, die gelöscht wurden.

```
tux> # In session 1.
tux> gstat test -r -t NORMAN

...
Analyzing database pages ...
NORMAN (142)
  Primary pointer page: 268, Index root page: 269
  Average record length: 9.00, total records: 15
  Average version length: 0.00, total versions: 0, max versions: 0
  Data pages: 1, data page slots: 1, average fill: 10%

tux> isql tset -user norman -password secret
Database: employee

SQL> SELECT COUNT(*) FROM NORMAN;

      COUNT
=====
          15
```

Zu diesem Zeitpunkt können wir sehen, dass die NORMAN-Tabelle 15 Datensätze enthält und dass die durchschnittliche Länge dieser 15 Datensätze 9,00 Byte beträgt. Als nächstes starten wir eine weitere isql-Sitzung und löschen alle Datensätze aus der NORMAN-Tabelle.

```
tux> # In session 2.
tux> isql test -user norman -password secret
Database: employee

SQL> DELETE FROM NORMAN;
SQL> COMMIT;
SQL> shell;
```

Noch in der zweiten Sitzung führen wir gstat aus, um Statistiken für die NORMAN-Tabelle abzurufen. Die Ergebnisse sind unten gezeigt.

```
tux> gstat test -r -t NORMAN

...
Analyzing database pages ...
NORMAN (142)
  Primary pointer page: 268, Index root page: 269
  Average record length: 0.00, total records: 15
  Average version length: 9.00, total versions: 15, max versions: 1
  Data pages: 1, data page slots: 1, average fill: 16%
...

tux> # Return to isql.
tux> exit
```

Wenn wir den obigen Bericht mit dem Bericht vergleichen, der vor dem Löschen der Datensätze erstellt wurde, können wir sofort feststellen, dass:

- Die durchschnittliche Datensatzlänge gibt an, dass die Tabelle keine Datensätze enthält, die Gesamtzahl der Datensätze zeigt jedoch, dass (noch) 15 vorhanden sind. Dies ist ein guter Indikator dafür, dass eine Sitzung alle Datensätze gelöscht hat, die Speicherbereinigung jedoch noch nicht ausgeführt wurde.
- Die Versionsdetails haben sich alle geändert. Es gibt jetzt Statistiken zur durchschnittlichen Versionslänge, Gesamtversionen und Maximalversionen.
- Die durchschnittliche Füllung für die Seite(n) in dieser Tabelle ist von 10% auf 16% gestiegen, obwohl alles gelöscht wurde. Der zusätzliche Speicherplatz wird von den früheren Versionen der gelöschten Datensätze verwendet.

Wenn wir in der zweiten Sitzung fortfahren und einen vollständigen Tabellenscan der NORMAN-Tabelle ausführen, werden keine Ergebnisse angezeigt, aber die Back-Versionen werden im Müll gesammelt.

```
SQL> SELECT * FROM NORMAN;

SQL> shell;

tux> gstat test -r -t NORMAN

...
Analyzing database pages ...
NORMAN (142)
  Primary pointer page: 268, Index root page: 269
  Average record length: 0.00, total records: 0
  Average version length: 0.00, total versions: 0, max versions: 0
  Data pages: 0, data page slots: 0, average fill: 0%
```

Alles ist jetzt auf Null zurückgekehrt. Es gibt keine früheren Versionen, keine aktuellen Versionen und die Seite ist nicht mehr gefüllt.

Average version length

Dies ähnelt der durchschnittlichen Datensatzlänge, gilt jedoch für die früheren Versionen des Datensatzes. Wenn Sie beispielsweise eine Reihe von Datensätzen gelöscht und andere aktualisiert haben, werden die alten — früheren — Versionen dieser Datensätze hier gemeldet. Wenn die Zahl 0,00 ist, hat die Speicherbereinigung stattgefunden und die früheren Versionen entfernt — siehe oben für ein Beispiel.

Total versions

Entspricht den oben genannten Gesamtdatensätzen, enthält jedoch nur die früheren Versionen. Wenn die Zahl 0 ist, hat die Speicherbereinigung stattgefunden und die früheren Versionen entfernt — siehe oben für ein Beispiel.

Max versions

Wenn ein Datensatz mehrmals aktualisiert wurde, zeigt die Statistik der maximalen Versionen die Anzahl der früheren Versionen des betreffenden Datensatzes (oder der betreffenden Datensätze) an. In einer Tabelle, in der alle Zeilen siebenmal aktualisiert wurden, eine jedoch 20mal aktualisiert wurde, gibt diese Statistik den Wert 20 aus. Wenn die Zahl 0,00 ist, hat die Speicherbereinigung stattgefunden und die früheren Versionen entfernt — siehe oben für ein Beispiel.

3.8. Wenn Sie eine beschädigte Datenbank haben

Im unwahrscheinlichen Fall einer Datenbankbeschädigung kann Ihre gstat -Ausgabe im Bericht Folgendes enthalten:

```
Database file sequence:
File /opt/firebird/examples/empbuild/corrupt.fdb is the only file

Analyzing database pages ...
    Expected b-tree bucket on page 337334 from 146314
```

Wenn Sie jemals eine Meldung wie die oben genannte sehen, die direkt nach den Header-Informationen angezeigt wird, sollten Sie sofort alle Verbindungen zur Datenbank beenden, eine Kopie der Datenbankdatei(en) auf Betriebssystemebene erstellen und versuchen, gbak auszuführen `gegen die Datenbank, um eine vollständige Sicherung zu erstellen. Die Verwendung von `nbackup kopiert die Datenbank möglicherweise problemlos, meldet jedoch keine Fehler. gbak hingegen meldet Fehler.

Kapitel 4. gstat-Fallstricke

Das Folgende ist eine kurze Liste von Fallstricken und Funnies, die ich bei meiner eigenen Verwendung von gstat entdeckt habe. Einige davon sind oben erwähnt, andere möglicherweise nicht. Wenn Sie sie alle hier an einem Ort sammeln, sollten Sie herausfinden können, was passiert, wenn Sie Probleme haben.

4.1. Der Schalter `-t[able]` kann Probleme machen

Der Schalter `-t[able]` erwartet, dass eine Liste von Tabellennamen (in Großbuchstaben) angegeben wird. Wenn Sie den Datenbanknamen *nach* einem Tabellennamen angeben, wird dieser leider als Tabellennamen angenommen und Sie werden zur Eingabe eines Datenbanknamens aufgefordert.

```
tux> gstat -t EMPLOYEE JOB employee
please retry, giving a database name
```

Rufen Sie aus diesem Grund immer gstat mit dem Datenbanknamen als *ersten* Parameter auf:

```
tux> gstat employee -t EMPLOYEE JOB

Database "/opt/firebird/examples/empbuild/employee.fdb"
Database header page information:
...

Database file sequence:
File /opt/firebird/examples/empbuild/employee.fdb is the only file

Analyzing database pages ...
...
```

Alternativ können Sie einen zusätzlichen Schalter *nach* dem letzten Tabellennamen und *vor* dem Datenbanknamen angeben:


```
tux> gstat -t EMPLOYEE JOB -z employee
gstat version LI-V2.1.3.18185 Firebird 2.1

Database "/opt/firebird/examples/empbuild/employee.fdb"
Database header page information:
...

Database file sequence:
File /opt/firebird/examples/empbuild/employee.fdb is the only file
    Firebird/linux Intel (access method), version
"LI-V2.1.3.18185 Firebird 2.1"
    Firebird/linux Intel (remote server), version
"LI-V2.1.3.18185 Firebird 2.1/tcp (greenbird)/P11"
    Firebird/linux Intel (remote interface), version
"LI-V2.1.3.18185 Firebird 2.1/tcp (greenbird)/P11"
    on disk structure version 11.1

Analyzing database pages ...
```

4.2. Die Schattenzahl scheint falsch zu sein

Es scheint, dass das Hinzufügen und/oder Löschen von Schattendateien aus einer Datenbank nicht immer von gstat gemeldet wird, wenn ein Datenbankbericht erstellt wird.

```
tux> # Use gstat to display shadow details
tux> gstat employee -h|grep -i sh[a]dow
    Shadow count          0

tux> isql employee
Database: employee

SQL> SHOW DATABASE;
Database: employee
    Owner: SYSDBA
    Shadow 1: "/u00/firebird/databases/employee.shd1" auto
...
```

Es ist sofort offensichtlich, dass der Bericht von gstat falsch ist, da die Mitarbeiterdatenbank eine Schattendatei enthält. Wenn wir isql verwenden, um dieser Datenbank eine neue Schattendatei hinzuzufügen, wie unten gezeigt, besteht gstat weiterhin darauf, dass keine Schatten vorhanden sind.

```
SQL> CREATE SHADOW 7 AUTO '/u00/firebird/databases/employee.shd7';
```

```
SQL> SHOW DATABASE;
```

```
Database: employee
```

```
Owner: SYSDBA
```

```
Shadow 1: "/u00/firebird/databases/employee.shd1" auto
```

```
Shadow 7: "/u00/firebird/databases/employee.shd7" auto
```

```
...
```

```
SQL> shell;
```

```
tux> gstat employee -h | grep -i sh[a]dow
```

```
Shadow count          0
```

Anhang A: Dokumentenhistorie

Der genaue Dateiversionsverlauf wird im Firebird-Dokumentations-Git-Repository aufgezeichnet; siehe <https://github.com/FirebirdSQL/firebird-documentation>

Revisionshistorie

1.0	29. Okt. 2009	ND	Neues gstat-Handbuch erstellt.
1.1	30. Nov. 2009	ND	Viele von Paul Vinkenhoog vorgeschlagene Korrekturen sowie eine allgemeine Aufräumaktion und einige weitere Beispiele wurden hinzugefügt.
1.2	14. Dez. 2009	ND	Ein paar kleinere Korrekturen und Rechtschreibfehler wurden korrigiert.
1.3	17. Feb. 2010	ND	Formatierungsfehler in den Befehlszeilenschaltern wurden korrigiert.
1.4	23. Mär. 2011	ND	ODS 9.1 für Interbase 5.0 wurde zur Liste der bekannten ODS-Werte hinzugefügt. Verweis auf Verwalten von Limbo-Transaktionen im gfix-Handbuch hinzugefügt. Korrigierte Erklärung, wann ein automatischer Datenbank-Sweep durchgeführt wird, basierend auf OIT und OST im Gegensatz zu OIT und OAT. Wie von Vlad Khorsun empfohlen.
1.5	11. Okt. 2011	ND	Aktualisiert für Firebird 2.5. Rechtschreibfehler korrigiert.
1.6	19. Jun. 2020	M R	Konvertierung in AsciiDoc, geringfügige Bearbeitung von Texten
1.6- de	31. Jul. 2020	M K	Deutsche Übersetzung.

Anhang B: Lizenzhinweis

Der Inhalt dieser Dokumentation unterliegt der "Public Documentation License Version 1.0" (der "License"); die Dokumentation darf nur unter Respektierung dieser Lizenz genutzt werden. Kopien der Lizenz sind verfügbar unter <https://www.firebirdsql.org/pdfmanual/pdl.pdf> (PDF) und <https://www.firebirdsql.org/manual/pdl.html> (HTML).

Die Original-Dokumentation trägt den Titel *Firebird Database Statistics Reporting Tool*.

Der ursprüngliche Autor der Original-Dokumentation ist: Norman Dunbar.

Copyright © 2005-2020. Alle Rechte vorbehalten. Kontakt zum Original-Autor: NormanDunbar at users dot sourceforge dot net.

Mitwirkende: Norman Dunbar; Mark Rotteveel; Martin Köditz - siehe [Dokumenthistorie](#).