

Willkommen zum „IBM Informix Newsletter“

Inhaltsverzeichnis

Aktuelles.....	1
TechTipp: Informix und NoSQL (Teil 1): Terminologie, JSON und BSON.....	2
TechTipp: Datablades ohne Blademanager (bladmgr) registrieren.....	5
TechTipp: IBM SP(a)SS Statistics mit INFORMIX.....	7
TechTipp: Tenant databases.....	12
TechTipp: Tenant Databases im Open Admin Tool (OAT).....	14
TechTipp: Minimum „free space“ in ROOTDBS für Upgrade.....	15
Versionsinfo: 11.70.xC8W1 ist verfügbar.....	15
Anmeldung / Abmeldung / Anmerkung.....	15
Die Autoren dieser Ausgabe.....	16

Aktuelles

Liebe Leserinnen und Leser,

nun mitten in den Sommerferien häufen sich die Feste und Feuerwerke am See. Die Redaktion ist schon fast im Urlaub, hat für Sie aber noch als Urlaubslektüre diese Ausgabe zusammengestellt, damit Sie in aller Ruhe einmal über neue Dinge nachdenken können. NoSQL, Jeder spricht davon, aber „Was ist das eigentlich und wie funktioniert es?“ Dies ist der Start einer neuen Serie im Newsletter.

Zudem haben wir uns im Bereich der Frontends umgesehen und wollen Ihnen SPSS näher bringen. Wer die einfache Administration von INFORMIX mag, der wird sich schnell mit SPSS anfreunden können.



Wie immer haben wir für Sie eine Reihe an Tipps und Tricks zusammengestellt. Viel Spaß mit den Tipps der aktuellen Ausgabe.

Ihr TechTeam

TechTipp: Informix und NoSQL (Teil 1): Terminologie, JSON und BSON

Seit Informix 12.10.xC2 gibt es die NoSQL-Funktionalität, und sie wird mit jeder neuen Version erweitert. Eine Besonderheit von Informix ist hierbei, dass SQL-Daten und NoSQL-Daten zusammen verarbeitet werden können. Damit ist es an der Zeit, im Newsletter detaillierter darauf einzugehen. Das Thema ist nicht nur neu, sondern auch umfangreich, so dass wir es in einer Serie von Artikeln schrittweise angehen. Zum Auftakt beginnen wir in dieser Ausgabe daher mit allgemeiner Information zur Terminologie, zu den Datenformaten JSON und BSON und konzeptionellen Unterschieden zu SQL.

Terminologie und Konzept

Ob der Begriff *NoSQL* nun für 'No SQL' oder für 'Not Only SQL' steht, mag Ansichtssache sein. Fest steht, dass es Unterschiede zum relationalen Datenmodell und SQL gibt, beginnend mit der Terminologie. Einige der wichtigsten Begriffe und das zugrundeliegende Konzept schauen wir deshalb genauer an, um eine Basis für den Einstieg in die NoSQL-Welt zu haben.

Zwar gibt es auch bei NoSQL das Konzept einer *Datenbank* als logischen Behälter für verschiedene Daten, vergleichbar mit einer SQL-Datenbank. Hiermit ist die Gemeinsamkeit in der Terminologie auch schon so ziemlich erschöpft. Eine SQL-*Tabelle* ist ein recht starres Gebilde mit einer festen Anzahl von *Spalten*, worin die einzelnen Datensätze in *Zeilen* (rows) angeordnet sind. Die Entsprechung einer SQL-Tabelle ist bei NoSQL eine *Collection* (Sammlung). Eine *Collection* enthält *Dokumente*, welche den zeilenweisen Datensätzen einer SQL-Tabelle entsprechen. Im Gegensatz zur SQL-Tabelle mit der festen Anzahl von Spalten sind die *Dokumente* einer *Collection* nicht an ein festes Format gebunden. Vielmehr enthält ein *Dokument* sogenannte *Key-Value*-Paare, die nicht für alle *Dokumente* einer *Collection* gleichermassen festgelegt sind. Jedes individuelle *Dokument* kann verschiedene *key-value*-Paare enthalten, die sich nicht nur im Wert (value), sondern auch im Schlüssel (key) unterscheiden können. Daher wird NoSQL oft auch als *schemalos* und somit viel flexibler (als ein SQL-Schema) bezeichnet, weil *Dokumente* in einer *Collection* eben nicht dem Schema einer Tabelle mit einer definierten Anzahl von Spalten entsprechen.

Zum Beispiel kann eine *Collection* "Kunde" die folgenden zwei unterschiedlich strukturierten *Dokumente* enthalten:

Dokument 1:

Vorname: Hans
Nachname: Mustermann
Strasse: Kryddgårdsvägen 12
PLZ: 13754
Ort: Tungalsta
Land: Schweden

Dokument 2:

Firmenname: NoSQL-Experts
Postfach: 42
PLZ: 85739
Ort: Garching

Würden wir diese zwei *Dokumente* als Zeilen in eine SQL-Tabelle abspeichern, so müsste die Tabelle folgendermassen aussehen:

Vorname	Nachname	Firmen name	Strasse	Postfach	PLZ	Ort	Land
Hans	Mustermann		Kryddgårds vägen 12		13754	Tungelsta	Schweden
		NoSQL - Experts		42	85739	Garching	

In die *Collection* "Kunde", welche die zwei obigen *Dokumente* schon enthält, können wir ohne Weiteres ein drittes *Dokument* einfügen:

Dokument 3:

Organisation: UNO
 Niederlassung: Bonn
 Strasse: Platz der Vereinten Nationen 1
 PLZ: 53113
 Ort: Bonn
 Telefon: 0228 815 2773

Um den entsprechenden Datensatz in die SQL-Tabelle einzufügen, müssten wir zuerst die Tabellenstruktur um die neuen Spalten "Organisation", "Niederlassung" und "Telefon" erweitern. Dies ist bei der *Collection* nicht erforderlich, da es von vornherein kein Schema gibt.

JSON

JSON steht für 'JavaScript Object Notation', ein offener Standard zum Austausch von Daten als *key-value*-Paare in einem lesbaren Format. Obiges "Dokument 1" sieht im JSON-Format wie folgt aus:

```
{
  "Vorname": "Hans",
  "Nachname": "Mustermann",
  "Strasse": "Kryddgårdsvägen 12",
  "PLZ": 13754,
  "Ort": "Tungelsta",
  "Land": "Schweden"
}
```

JSON selbst ignoriert 'whitespace' ausserhalb der String-Werte. D.h. die obige Formatierung mit Leerzeichen und Zeilenumbrüchen verbessert die Lesbarkeit hier im Newsletter, ist für JSON aber nicht relevant. Zudem sind als solche 'whitespace'-Zeichen nur die folgenden vier erlaubt: Leerzeichen, horizontaler Tabulator, 'line feed' und 'carriage return'. Des weiteren gibt es keine Möglichkeit, Kommentare einzufügen, die bei der Verarbeitung zu ignorieren wären.

Dokumente im JSON-Format können auch geschachtelt sein und Arrays enthalten. Arrays werden mit den eckigen Klammern [] beschrieben. Wenn wir obiges "Dokument 1" etwas anders strukturieren, um die Adresse als (inneres) Dokument zusammenzufassen, und zusätzlich noch mehrere Telefonnummern in einem Array hinzufügen, dann könnte das Dokument in JSON-Notation so aussehen:

```
{
  "Vorname": "Hans",
  "Nachname": "Mustermann",
  "Adresse": {
    "Strasse": "Kryddgårdsvägen",
    "Hausnummer": "12",
    "PLZ": 13754,
    "Ort": "Tungelsta",
    "Land": "Schweden"
  },
  "Telefon": [
    { "Typ": "Festnetz", "Nummer": "+46 013 2785" },
    { "Typ": "Mobil", "Nummer": "+46 486 319275" }
  ]
}
```

BSON

Im Zusammenhang mit JSON taucht auch immer wieder der Begriff *BSON* auf, was für 'Binary JSON' steht. Wie der Name schon vermuten lässt, ist dies ein Format zur binären Codierung von JSON-Daten. BSON wird hauptsächlich mit der MongoDB Datenbank benutzt zur Speicherung und zum Datenaustausch über Programmschnittstellen hinweg. Hierfür ist BSON generell besser geeignet, da es maschinell effizienter verarbeitet werden kann. BSON wird intern auch für die NoSQL-Funktionalität von Informix genutzt, da es aber nicht lesbar ist, werden wir uns vorerst nicht im Detail damit beschäftigen.

Im nächsten Newsletter richten wir für eine Informix-Instanz die NoSQL-Funktionalität ein.

TechTipp: Datablades ohne Blademanager (blademgr) registrieren

Die Strukturen und Funktionen vieler Datablades werden in den aktuellen Versionen automatisch bei der ersten Nutzung registriert. Bei einigen Datablades ist weiterhin die explizite Registrierung in der Datenbank notwendig. Dies kann mittels Blade Manager (blademgr) erfolgen, oder mittels SQL-Aufrufen. Die Variante der Registrierung von Datablades mittels SQL Funktionen, die speziell für Datenbanken im Mode ANSI notwendig ist, wollen wir in diesem Artikel vorstellen.

Zuerst muss eine temporäre Struktur erstellt werden, welche die Registrierung vornimmt und protokolliert. Dies erfolgt mittels:

```
EXECUTE FUNCTION SYSBldPrepare('ifxmng', 'sysblderrorlog');
```

In der Tabelle sysblderrorlog werden die Meldungen zur Registrierung gesammelt. Die Tabelle kann mit normalem SQL abgefragt werden, z.B.:

```
SELECT *  
FROM 'informix'.sysblderrorlog  
WHERE err_expected != 'expected';
```

Die Where-Clause dient hier dazu übliche Meldungen von Fehlern zu unterscheiden, wie z.B. „dass eine ältere Version eines Datablades nicht gelöscht werden konnte, weil sie nicht installiert war“.

Um zu ermitteln welche Module registriert werden können, kann folgende Abfrage genutzt werden:

```
SELECT * FROM 'informix'.sysbldmodules;
```

```
bld_id           TimeSeries.6.00.FC4  
bld_server_type any  
bld_inst_state  0  
  
bld_id           bts.3.10  
bld_server_type any  
bld_inst_state  0  
  
bld_id           binaryudt.1.0  
bld_server_type any  
bld_inst_state  0  
  
bld_id           spatial.8.21.FC7  
bld_server_type any  
bld_inst_state  0
```

Um zu sehen welche Datablades bereits in der Datenbank registriert sind, kann folgender SQL abgesetzt werden:

```
SELECT * FROM 'informix'.sysbldregistered;
bld_id
ifxrltree.2.00
wfs.1.00.FC1
excompat.1.0
spatial.8.21.FC7
```

Nun wollen wir das Datablade „bts.3.10“ in der Datenbank registrieren:

```
EXECUTE FUNCTION SysBldRegister('bts.3.10', 'sysblderrorlog');
```

Einige Datablades erfordern als Voraussetzung das Vorhandensein anderer Datablades. So ist z.B. für die Volltextsuche mit Excalibur folgende Reihenfolge notwendig:

```
EXECUTE FUNCTION
SysBldRegister('LLD.1.20.UC2', 'sysblderrorlog');
EXECUTE FUNCTION
SysBldRegister('TXT.1.10.UC5', 'sysblderrorlog');
EXECUTE FUNCTION
SysBldRegister('ETX.1.30.UC5', 'sysblderrorlog');
```

Datablades können mittels „Unregister“ wieder aus der Datenbank entfernt werden:

```
EXECUTE FUNCTION
SysBldUnregister('bts.3.10', 'sysblderrorlog');
```

Um ein Datablade registrieren zu können, muss der User das Recht „Extend Role“ haben. Ist dieses Recht nicht vorhanden, so scheitert die Registrierung mit einem Returncode ungleich 0. In der Tabelle sysblderrorlog sind dann die Details zum Fehler zu finden:

```
order          3784
bld_id         bts.3.10
err_operation  SYSBldRegister
err_expected   unexpected
err_sql_stmt   create function bts_beginscan (pointer) |
returns
              integer|external

name "$INFORMIXDIR/extend/bts.3.10/bts.bld(bts_beginscan)"
language c;

err_sql_state  XIX000:-25785 : Cannot create external routine
(bts_beginscan) without the EXTEND role.
```

TechTipp: IBM SP(a)SS Statistics mit INFORMIX

Eine Datenbank lebt von den Zugriffen der Frontends. Die Produktfamilie IBM SPSS bietet hier interessante Möglichkeiten, die Daten der Datenbank zu analysieren.

Zu IBM SPSS gehören:

IBM SPSS Statistics

Prognostizieren: Schnelle und einfache Erstellung von zukunftsweisenden, statistischen Analysen, Identifikation von Trends und Generierung von präzisen Forecasts.

IBM Analytical Decision Management

Handeln: Automatisierte Zurverfügungstellung von optimierten Entscheidungshilfen für Frontline-Systeme oder Entscheidungsträger.

IBM SPSS Data Collection

Erfassen: Genauere Einsicht in Einstellungen, Vorlieben und Meinungen von Personen.

IBM SPSS Modeler

Vorhersagen: Entdecken von Mustern und Trends in Daten mithilfe einer einzigartigen, intuitiven, visuellen Oberfläche sowie fortgeschrittenen Analysetechniken.

Kundenanalyse

Aufdecken von verborgenen Erkenntnissen aus Kundendaten, Erstellen von personalisierten Angeboten, die mehr Umsatz generieren bei gleichzeitiger Kostenreduzierung und Steigerung der Kundentreue.

Sowie weitere Analyselösungen

An dieser Stellen möchten wir Ihnen das Zusammenspiel zwischen **IBM SPSS Statistics** und der Informix Datenbank näher vorstellen.

Für unseren Test haben wir das Programm „SPSS_Statistics_22_lin“ mit allen verfügbaren Fixes installiert. Die Installation wird analog anderer IBM Produkte über ein Menu geführt. Für den Zugriff auf die Datenbank ist der ODBC-Treiber von DataDirect notwendig, der ebenfalls installiert werden muss.

Für das Arbeiten mit SPSS muss der LD_LIBRARY_PATH und der PATH um die Komponenten des SPSS erweitert werden. Im Beispiel:

```
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/opt/IBM/SPSS/Statistics/22/lib
export PATH=$PATH:/opt/IBM/SPSS/Statistics/22/bin
```

In der Datei „statsenv.sh“ (im \$SPSS/bin) muss noch auf das Script verwiesen werden, das die Umgebungen für ODBC setzt:

```
MERANT_ENVIRONMENT_SCRIPT=/opt/IBM/SDAP71/odbc.sh
```

Im Script „odbc.sh“ sind folgende Variablen wichtig:

```
export ODBCINI=/opt/IBM/SDAP71/odbc.ini  
export ODBCINST=/opt/IBM/SDAP71/odbcinst.ini
```

Nun fehlen nur noch die Einträge für INFORMIX in der Datei „odbc.ini“.

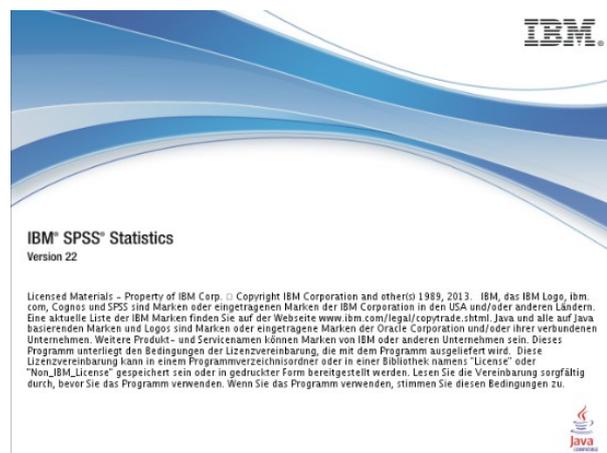
Bei der Liste der Treiber wird eine zusätzliche Zeile eingefügt:

```
IFX_STORES_SPSS=IBM Corp. 7.1 Informix Wire Protocol
```

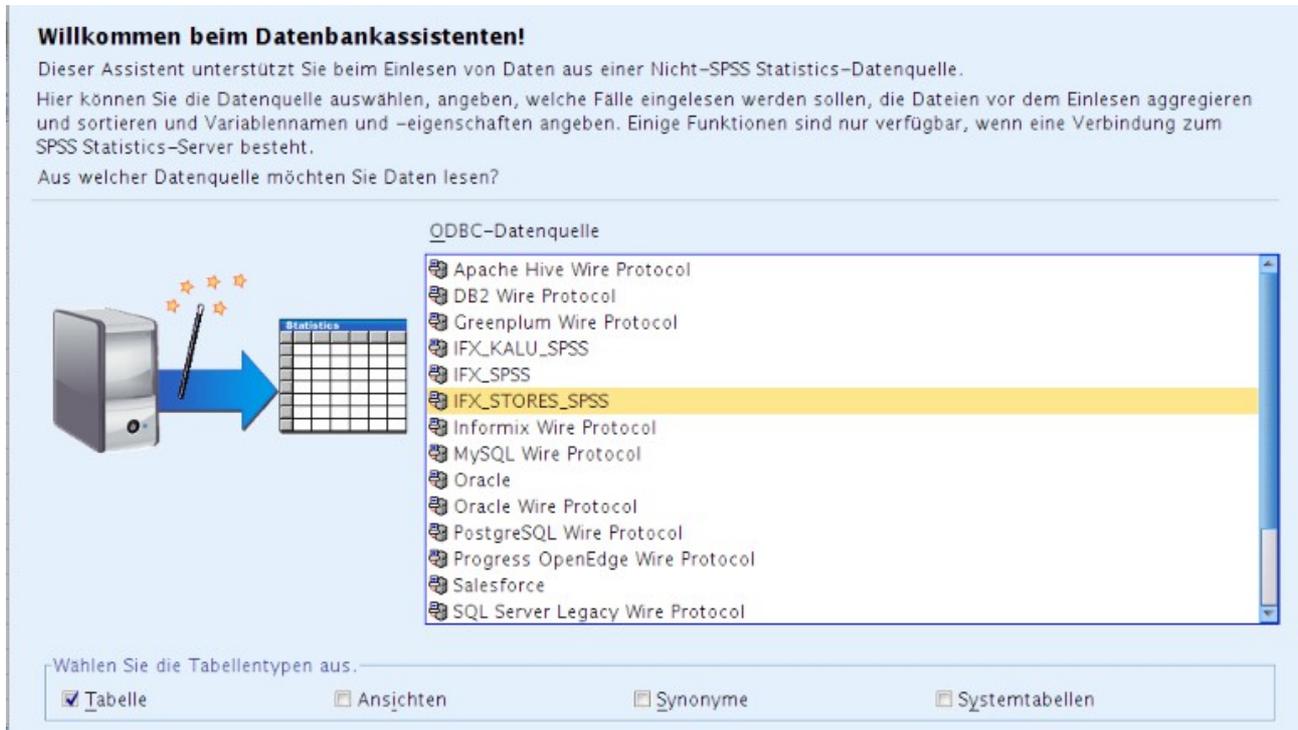
zu der dann weiter unten ein Abschnitte erstellt wird (Kopie einer anderen Sektion) die Details aufgelistet werden (Notwendige Anpassungen in **ROT**):

```
[IFX_STORES_SPSS]  
Driver=/opt/IBM/SDAP71/lib/XEifc127.so  
Description=IBM Corp. 7.1 Informix Wire Protocol  
AlternateServers=  
ApplicationUsingThreads=1  
CancelDetectInterval=0  
ConnectionRetryCount=0  
ConnectionRetryDelay=3  
Database=stores  
HostName=172.16.41.230  
LoadBalancing=0  
LogonID=  
Password=  
PortNumber=9088  
ServerName=ifxibm  
TrimBlankFromIndexName=1  
UseDelimitedIdentifiers=0
```

Nach diesen Vorbereitungen kann SPSS mit dem Befehl „stats“ gestartet werden.



Mit dem Aufruf **Datei → Datenbank öffnen → Neue Abfrage** gelangt man zum Datenbankassistenten, der alle eingetragenen ODBC-Verbindungen auflistet:



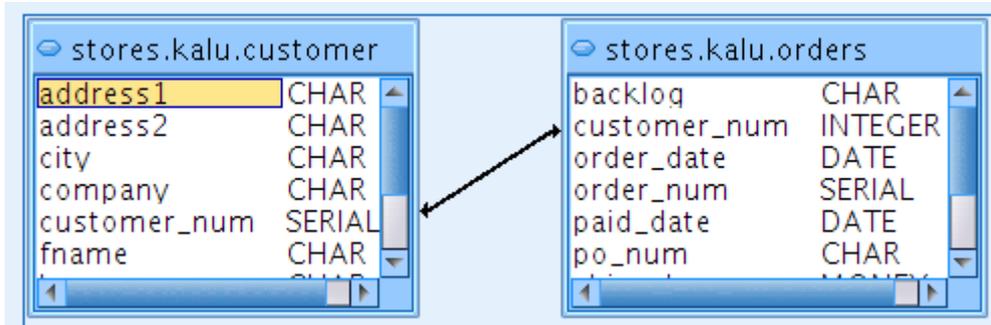
Die zur Verfügung stehenden Tabellen werden aufgelistet. Die Felder für die Datenbankabfragen können nun mittels Drag&Drop ausgewählt werden:



Achtung:

Nur die Tabellen des Schemas (Owner), mit dem der Connect aufgebaut wird, werden angezeigt !

Werden Felder mehrerer Tabellen ausgewählt, so müssen anschliessend noch die Verknüpfungen zwischen den Tabellen explizit angegeben werden, indem die Join-Felder ebenfalls mittels Drag&Drop aufeinander gezogen werden:



Sind diese Schritte erfolgt, so wird nach der Auswahl „Fertigstellen“ die Tabelle der ausgewählten Daten angezeigt.

Nun kann man im Menüpunkt „Transformieren“ unter „Visuelle Klassifizierung“ noch eine Einteilung in Klassen vornehmen (im Beispiel wurden Unterteilungen von 10% gewählt). Im Menüpunkt „Graphik“ kann nun die gewünschte Grafik ausgewählt werden, die Beschriftungen der Achsen bestimmt und eine Überschrift gewählt werden.

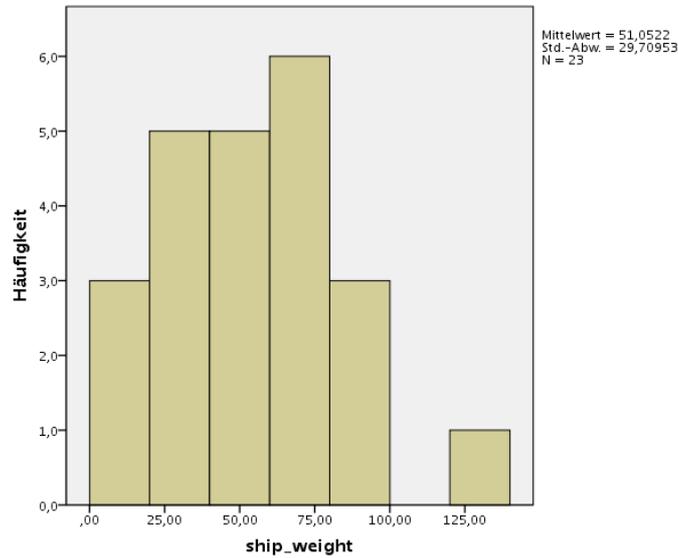
Sind diese Schritte getan, so erstellt das Drücken des grünen Pfeils die Ausgabe, wobei zusätzlich die Daten der Verbindung, die SQL-Abfrage und die Parameter der gewählten Grafik angezeigt werden:

```
GET DATA
  /TYPE=ODBC
  /CONNECT='DSN=IFX_STORES_SPSS;UID=kalu;PWD==J#E,!-!=c#{/!-,6N/~;DB=stores;HOST='+
  '172.16.41.230;Port=9088;SRVR=ifxibm'
  /SQL='SELECT customer_num, ship_weight FROM kalu.orders'
  /ASSUMEDSTRWIDTH=255.

CACHE.
EXECUTE.
DATASET NAME DataSet2 WINDOW=FRONT.
EXECUTE.
* Diagrammerstellung.
GGRAPH
  /GRAPHDATASET NAME="graphdataset" VARIABLES=ship_weight MISSING=LISTWISE REPORTMISSING=NO
  /GRAPHSPEC SOURCE=INLINE.
BEGIN GPL
  SOURCE: s=userSource(id("graphdataset"))
  DATA: ship_weight=col(source(s), name("ship_weight"))
  GUIDE: axis(dim(1), label("ship_weight"))
  GUIDE: axis(dim(2), label("Häufigkeit"))
  ELEMENT: interval(position(summary.count(bin.rect(ship_weight))), shape.interior(shape.square))
END GPL.
```

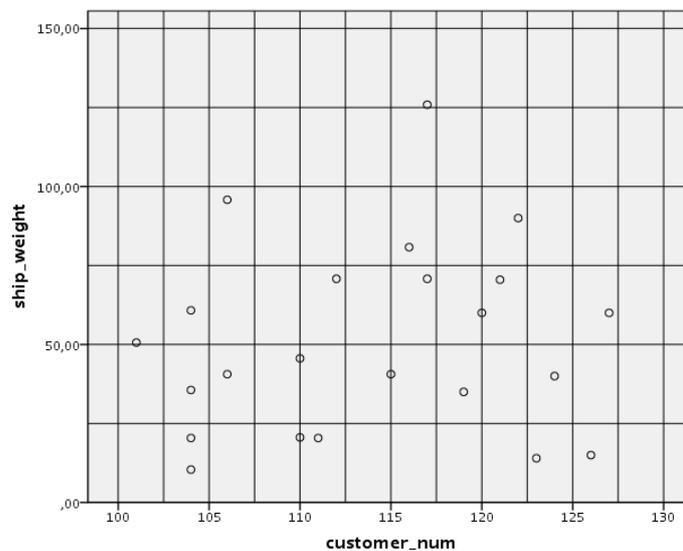
Das in diesen einfachen Schritten erstellte Chart sieht nun folgendermassen aus:

ShipWeight Verteilung



Analog lässt sich die Werteverteilung darstellen:

➔ Ship weight for customers



SPSS Statistics ist ein sehr mächtiges Tool, mit dem man in kurzer Zeit statistische Auswertungen erstellen, Vorhersagen treffen, Analysen und Verifizierungen vornehmen kann. Mit wenigen Schritten ist anschliessend die Umsetzung dieser Auswertungen in die gewünschte Graphik möglich.

SPSS Statistics ist offiziell auf den Plattformen

- Linux
 - Debian 6.0 (x86_64)
 - RedHat Enterprise 5 (x86_64)
 - RedHat Enterprise Client 6 (x86_64)
- MacOS
 - OS X Lion 10.7 (x86-64)
 - OS X Mountain Lion 10.8 (x86-64)
 - OS X Mavericks 10.9 (x86-64)
- Windows
 - x86_32 und x86_64
 - von WindowsXP bis Windows 8.1

unterstützt.

Unsere Tests fanden auf Ubuntu 12.04 (x86_64) statt, wobei es keinerlei Probleme bei der Installation oder den Abfragen gab.

TechTipp: Tenant databases

Unter „Multi Tenancy“ verbirgt sich die Zuteilung von Ressourcen für unterschiedliche Datenbanken innerhalb einer Instanz. Hierbei werden neben den Daten auch die Storagebereiche (DBSpaces/Chunks) und die virtuellen Prozessoren exklusiv zugeteilt. Bei den temporären Bereichen besteht die Option diese getrennt, oder gemeinsam zu nutzen.

Zum besseren Verständnis, hilft ggf. die Vorstellung, dass ein Tenant einen Mandanten darstellt, der gewisse Ressourcen und Rechte gebucht hat, aber nicht die Ressourcen der anderen Mandanten benutzen darf.

Damit können Ressourcen individuell zugeteilt werden, und es müssen keine eigenständigen Instanzen je Mandant erstellt werden.

Eine Tenant Datenbank wird nicht mittels „create database“ erstellt, sondern durch die Ausführung eines Tasks in der Datenbank „sysadmin“.

Bei der Anlage der Datenbank wird angegeben, welche DBSpaces von dieser Datenbank genutzt werden dürfen, welcher DBSpaceTemp und welcher SBSpace verfügbar sein sollen. Ebenso wird der Name der virtuellen Prozessoren, die für diese Datenbank zur Verfügung stehen, angegeben (diese müssen zuvor mittels „onmode -p“ gestartet sein, bzw. sie müssen über VPCLASS in der Konfiguration festgelegt sein). Auch die maximale Anzahl von Sperrern je Session, der Loggingmode der Datenbank und die zu nutzende DB_LOCALE werden bei der Erstellung mit angegeben.

Als weitere Option besteht die Möglichkeit, die Datenbank CaseSensitiv anzulegen. Der Default ist 'Case Insensitiv'.

Beispiel für die Erstellung einer Tenant Datenbank „kalu_ten“:

```
EXECUTE FUNCTION task('tenant create', 'kalu_ten',
  '{dbspace:"kalu_dbs1, kalu_dbs2",
   sbspace:"kalu_sbs",
   vpclass:"kalu_vp,num=2",
   dbspacetemp:"kalu_tempdbs",
   session_limit_locks:"42000",
   logmode:"UNBUFFERED",
   locale:"en_us.utf8"}'
);
```

Result:

```
(expression) Tenant kalu_ten successfully added.
```

Wird versucht aus der Datenbank heraus auf einen Datenbereich zuzugreifen, der nicht diesem Mandanten zugeordnet ist, so wird die Fehlermeldung „No user permission“ ausgegeben:

Beispiel:

```
create table test (nr int) in datadbs
```

```
9662: No user permission for datadbs
```

Die Ausgabe von „onstat -g ath“ zeigt, dass die Threads, die auf dieser Datenbank arbeiten, auf den hierfür zur Verfügung gestellten VPs laufen:

```
77 4eeb56c8 4653e228 1 running 10kalu_vp sqlexec
```

Auch der „onstat -g glo“ zeigt, dass die Last auf den vorgesehenen VPs abgearbeitet wird:

Virtual processor summary:

class	vps	usercpu	syscpu	total
cpu	1	3.62	0.74	4.36
aio	2	0.04	0.29	0.33
...				
msc	1	0.00	0.00	0.00
fifo	1	0.00	0.00	0.00
kalu_vp	2	46.42	8.74	55.16
total	11	51.27	12.35	63.62

Eine Tenant Database kann nicht mittels „drop database“ gelöscht werden. Beim Versuch kommt die Meldung:

9663: Cannot Drop tenant database

Das Löschen muss, wie das Erstellen, über den Aufruf eines Tasks erfolgen:

```
execute function task ('tenant drop','kalu_ten')
```

Achtung ! ... folgender Befehl löscht die Datenbank NICHT vollständig !!!

```
execute function task ('drop database','kalu_ten')
```

und sollte daher nicht ausgeführt werden !

TechTipp: Tenant Databases im Open Admin Tool (OAT)

Das OAT wurde im Bereich „Storage“ um den Anwahlpunkt „Tenant“ erweitert, unter dem sich die Ressourcen eines Tenants anzeigen lassen.

Beispiel:

Database	Tables	Indexes	Compressed	VPID	Name	Ready Queue	CPU Time
kalu_ten	65	112	0	8	kalu_vp.num=2	0	0.00

Item	Status	% Used	Size	Number	Space	Status	% Used	Size	Path
kalu_dbt1	●		97.66 MB	6	kalu_dbt1	●		97.66 MB	/kalu_dbt1
kalu_dbt2	●		97.66 MB	7	kalu_dbt2	●		97.66 MB	/kalu_dbt2
kalu_sbs	●		97.66 MB	8	kalu_sbs	●		97.66 MB	/kalu_sbs

TechTipp: Minimum „free space“ in ROOTDBS für Upgrade

In den Checklisten zur Migration wird explizit darauf hingewiesen, dass man darauf achten sollte, dass im ROOTDBS wenigstens 100 MB freier Platz zur Verfügung steht. Da wir bei Tests auf einer möglichst klein gehaltenen Instanz jedoch selbst in dieses Problem gelaufen sind, hier nochmals die Warnung, VOR einem Upgrade dafür zu sorgen, dass ausreichend Platz für die Migration im ROOTDBS vorhanden ist.

Sollte man dies vergessen, ist die Fehlermeldung beim Start mit der neuen Version immerhin eindeutig:

```
07/01/14 15:50:59 The minimum free space required for conversion in
ROOTDBS (rootdbs) is 90000K. Current free space is 33044K.
```

Versionsinfo: 11.70.xC8W1 ist verfügbar

Seit einigen Tagen ist die Version 11.70.xC8W1 für alle unterstützten Plattformen und Editionen verfügbar. Da es in jeder Version eine Reihe an Verbesserungen gibt, sollte immer eine der neueren Versionen eingesetzt werden.

Anmeldung / Abmeldung / Anmerkung

Der Newsletter wird ausschließlich an angemeldete Adressen verschickt. Die Anmeldung erfolgt, indem Sie eine Email mit dem Betreff „**ANMELDUNG**“ an **ifmxnews@de.ibm.com** senden.

Im Falle einer Abmeldung senden Sie „**ABMELDUNG**“ an diese Adresse.

Das Archiv der bisherigen Ausgaben finden Sie zum Beispiel unter:

<http://www.iug.org/intl/deu>

http://www.iug.de/index.php?option=com_content&task=view&id=95&Itemid=149

<http://www.informix-zone.com/informix-german-newsletter>

<http://www.drap.de/link/informix>

<http://www.nsi.de/informix/newsletter>

<http://www.cursor-distribution.de/index.php/aktuelles/informix-newsletter>

<http://www.listec.de/Newsletter/IBM-Informix-Newsletter/View-category.html>

<http://www.bereos.eu/software/informix/newsletter/>

Die hier veröffentlichten Tipps&Tricks erheben keinen Anspruch auf Vollständigkeit. Da uns weder Tippfehler noch Irrtümer fremd sind, bitten wir hier um Nachsicht falls sich bei der Recherche einmal etwas eingeschlichen hat, was nicht wie beschrieben funktioniert.

Die Autoren dieser Ausgabe

Gerd Kaluzinski IBM Informix Services, Europa
IBM Software Group Services, IT-Specialist Informix
gerd.kaluzinski@de.ibm.com +49-175-228-1983

Martin Fuerderer IBM Informix Entwicklung, München
IBM Software Group, Information Management
martinfu@de.ibm.com

Markus Holzbauer IBM Informix Advanced Support, München
IBM Software Group, Information Management Support
holzbauer@de.ibm.com

Christian Hiotu IBM Advanced Support, München
IBM Software Group, SPSS Senior Support Specialist

Die Versionsinfo stammt aus dem Versions-Newsletter der CURSOR Software AG
<http://www.cursor-distribution.de/download/informix-vinfo>

Sowie unterstützende Teams im Hintergrund.

Fotonachweis: Gerd Kaluzinski

(Seehasenfest FN 2014)