

Willkommen zum „IBM Informix Newsletter“

Inhaltsverzeichnis

Aktuelles.....	1
TechTipp: IWA 12.10 Feature: SQL-API zur Administration.....	2
TechTipp: IWA Data Mart Definitionen und deren Handhabung.....	5
TechTipp: ENVIRONMENT – IFX_BAR_USE_DEDUP (nur mit TSM).....	7
TechTipp: Alter table InPlace für Serial8.....	8
TechTipp: Pending „Alter Table“ ermitteln.....	8
TechTipp: LATERAL als Kennung für errechnete Tabellen.....	9
TechTipp: Unterstützung für .NET.....	9
TechTipp: RSS und SDS um TimeSeries erweitert.....	9
WebTipp: Smarter Cities – die Zukunft der Städte.....	10
Anmeldung / Abmeldung / Anmerkung.....	10
Die Autoren dieser Ausgabe.....	11

Aktuelles

Liebe Leserinnen und Leser,

nachdem der Winter nicht stattgefunden hat, meldet sich wenigstens das Frühjahr gleich mit Sonne und angenehmen Temperaturen.

Bei INFORMIX startet das Jahr bereits mit vielen interessanten Projekten und Herausforderungen. Die Präsentation von INFORMIX auf der CeBIT, die Planung der Workshops und Bootcamps für 2014 und viele weitere Aktivitäten stehen an. Hierbei rückt das Thema Smarter Cities immer mehr in den Fokus (siehe auch WebTipp).



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

Ihr TechTeam

TechTipp: IWA 12.10 Feature: SQL-API zur Administration

Bis zum Erscheinen der Version 12.10.xC1 stand für eine programmatisch orientierte Administration des Informix Warehouse Accelerators (IWA) nur das sog. Java Command Line Interface (Java CLI) zur Verfügung. Die Version 12.10.xC1 bietet nun als komfortables SQL-API einen Satz von Stored Procedures an, mit denen alle Operationen durchgeführt werden können. Die Handhabung der Ausgabe im XML-Format entfällt. Die Stored Procedures können manuell und einzeln (z.B. in dbaccess) ausgeführt werden, in SQL-Scripts integriert werden, oder von einer Applikation direkt benutzt werden.

Mit der neuen Version 12.10.xC2 wurde die SQL-API mit zusätzlichen Stored Procedures erweitert für die neuen Funktionalitäten zur Integration von TimeSeries Daten. Diese Stored Procedures sind in der Beschreibung gekennzeichnet.

Eine detaillierte Betrachtung der einzelnen Stored Procedures würde den Rahmen des Newsletter bei weitem sprengen. Deshalb finden Sie hier eine Kurzbeschreibung und wir bitten Sie, für weitergehende Informationen das Handbuch *IBM Informix Warehouse Accelerator Administration Guide*, Abschnitt *SQL administration routines* zu Rate zu ziehen, zu finden bei:

https://pic.dhe.ibm.com/infocenter/informix/v121/topic/com.ibm.acc.doc/ids_acc_073.htm

Die Stored Procedures des SQL-API sind entsprechend ihrer Aufgaben in verschiedene Gruppen unterteilt. Hierbei sind die Stored Procedures der ersten Gruppe die einzigen, die nicht mit einem Data Mart zusammenhängen, sondern der Verwaltung eines Accelerators dienen.

1. Stored Procedures zur Administration eines Accelerators:

- **ifx_setupDWA()** leitet die Verbindung eines Informix Servers mit IWA ein und erzeugt einen neuen *Accelerator*.
- **ifx_removeDWA()** entfernt einen existierenden Accelerator aus der sqlhosts-Datei des Informix Servers. Zur Beachtung: Dies entfernt weder existierende Data Marts noch deren Metadaten im Informix Server.
- **ifx_getdwaMetrics()** ruft einige Statistiken zu IWA ab.

2. Stored Procedures zum Erzeugen oder Entfernen eines Data Marts und zum Ändern des Status eines existierenden Data Marts:

- **ifx_createMart() Typ 1** erzeugt einen neuen Data Mart in einem Accelerator entsprechend der spezifizierten XML Definition.
- **ifx_createMart() Typ 2** erzeugt einen neuen Data Mart in einem Accelerator mittels einer internen, relationalen Darstellung einer Data Mart Definition (mehr dazu im folgenden TechTipp).
- **ifx_dropmart()** entfernt den spezifizierten Data Mart in einem Accelerator.
- **ifx_setMart()** aktiviert oder deaktiviert einen existierenden Data Mart.

3. Stored Procedures zur Verwaltung der Daten in einem existierenden Data Mart:

- **ifx_loadMart()** lädt die kompletten Daten in einen Data Mart (sog. "full load"). Dies ist in jedem Fall nötig, um einen neu erzeugten Data Mart mit Daten zu füllen, und sollte danach in regelmässigen Abständen wiederholt werden.
- **ifx_loadPartMart()** und **ifx_dropPartMart()** lädt bzw. löscht die Daten einer bestimmten Partition der angegebenen Tabelle. Hierzu muss der Name der Tabelle, und für eine fragmentierte Tabelle auch der Name oder die Partitionsnummer des Fragments angegeben werden.
- **ifx_refreshMart()** stellt automatisch fest, welche Partitionen in einem Data Mart geändert wurden. Dies schliesst Fragmente ein, die neu hinzugefügt oder entfernt wurden. Die Daten der so bestimmten Partitionen werden dann entsprechend entfernt und/oder neu geladen.
- **ifx_setupTrickleFeed()** und **ifx_removeTrickleFeed()** beginnen oder beenden das kontinuierliche Einpflegen von Daten, die zu den Faktentabellen hinzugefügt werden. Um die Performance zu verbessern, erfolgt dies in Intervallen, deren Abstand spezifiziert werden kann. Zusätzlich können mit jedem Intervall auch die Daten der Dimensionstabellen erneuert werden.
- Stored Procedures für die Benutzung von Timeseries Virtual Table Interface (VTI) Tabellen und die Erneuerung ihrer Daten im Data Mart. Diese Stored Procedures sind mit der Version 12.10.xC2 neu zum SQL-API hinzugekommen:
 - **ifx_TSDW_createWindow()** begrenzt die Daten einer TimeSeries VTI Tabelle auf ein bestimmtes Zeitfenster, dessen Beginn und Ende anzugeben ist. Die Daten werden dann in den Data Mart geladen.
 - **ifx_TSDW_dropWindow()** entfernt ein Zeifenster einer TimeSeries VTI Tabelle, das zuvor mit **ifx_TSDW_createWindow()** erzeugt wurde. Die Daten werden aus dem Data Mart gelöscht.
 - **ifx_TSDW_moveWindow()** verschiebt ein existierendes Zeitfenster einer TimeSeries VTI Tabelle. Alte Daten im Data Mart werden gelöscht und die neuen Daten werden geladen.
 - **ifx_TSDW_setCalender()** teilt einer TimeSeries VTI Tabelle einen TimeSeries Calendar zu.
 - **ifx_TSDW_updatePartition()** erneuert für eine TimeSeries VTI Tabelle die Daten im Data Mart entsprechend dem zuvor zugeteilten TimeSeries Calendar.

4. Stored Procedures, die Informationen zu existierenden Data Marts liefern:
 - **ifx_listMarts()** zeigt alle Data Marts des gegebenen Accelerators an (unabhängig davon, für welche Datenbank die Data Marts erzeugt wurden). Zusätzliche Information zu jedem Data Mart ist seine Größe und der Zeitpunkt des letzten Ladens oder der letzten Datenerneuerung.
 - **ifx_getMartStat()** zeigt den Status des spezifizierten Data Marts an.
 - **ifx_getMartdef()** liefert die komplette XML Definition für den gegebenen Data Mart.

5. Stored Procedures zur Handhabung von Data Mart Definitionen:
 - **ifx_probe2Mart()** erzeugt aus Informationen einer Workload-Analyse eine interne, relationale Darstellung der Data Mart Definition.
 - **ifx_genMartDef()** erzeugt aus der internen, relationalen Darstellung die zugehörige XML Data Mart Definition.
 - **ifx_xml2Mart()** erzeugt aus einer XML Data Mart Definition die zugehörige interne, relationale Darstellung.
 - **ifx_def2Mart()** erzeugt für einen existierenden Data Mart die interne, relationale Darstellung der Definition.

Die Handhabung der Informationen zur Data Mart Definition scheint auf den ersten Blick recht kompliziert zu sein. Das Konzept und dessen Basis der verschiedenen Arbeitsschritte wird im nächsten TechTipp näher erläutert.

Privilegien zur IWA Administration

Ab Version 12.10.xC1 des Informix Warehouse Accelerator (IWA) muss ein Benutzer die nötigen Privilegien haben, um IWA Administrationsaufgaben ausführen zu können. Der Benutzer muss entweder 'informix' sein, die Rolle eines DBA innehaben, oder das WAREHOUSE Privileg besitzen.

Das WAREHOUSE Privileg kann einem Benutzer mit den Funktionen `admin()` oder `task()` übertragen werden. Die Funktionsargumente sind hierbei 'grant admin' und 'WAREHOUSE'.

Beispiel zur Übertragung des WAREHOUSE Privilegs auf den Benutzer 'kalu':

```
EXECUTE FUNCTION task('grant admin', 'kalu', 'WAREHOUSE');
```

TechTipp: IWA Data Mart Definitionen und deren Handhabung

Das Konzept der IWA Data Mart Definitionen basiert auf den verschiedenen Arbeitsschritten, bei denen eine Data Mart Definition benutzt wird. Am Beispiel einer Workload Analyse wird das Konzept verständlich:

1. Durchführen einer Workload Analyse mit "query probing":

In einer Workload Analyse wird für einen bestimmten Zeitraum beobachtet, welche SQL-Befehle im Informix Server abgearbeitet werden. Die Gesamtheit dieser SQL-Befehle bildet den Workload. Für die Analyse wird für die relevanten SQL-Befehle geprüft, ob sie beschleunigt werden können (sog. "probing"). Die relevanten SQL-Befehle sind Abfragen (SELECT), die an die Zieldatenbank gerichtet sind. (Im Zusammenhang mit IWA sind andere SQL-Befehle, z.B. INSERT, UPDATE und DELETE, nicht interessant, da sie nicht beschleunigt werden können.) Für Abfragen, die beschleunigt werden können, wird die Information gespeichert, die später zum Erstellen der Data Mart Definition benötigt wird. Diese Information enthält die an der Abfrage beteiligten Tabellen, deren Spalten und Verknüpfung (joins). Die Speicherung erfolgt nur im SHM des Informix Servers (nicht permanent). Mit dem Befehl `onstat -g probe` kann man die gespeicherten Informationen ansehen, wobei die Ausgabe des Befehls etwas kryptisch ist, da nicht die Tabellen- und Spaltennamen, sondern nur die jeweiligen IDs (als Integer) gespeichert sind. Die Informationen bleiben gespeichert bis sie entweder explizit gelöscht werden, oder der Informix Server neu gestartet wird.

2. Erzeugen einer Data Mart Definition aus den Probing Daten:

Hierzu wird die Stored Procedure `ifx_probe2Mart()` ausgeführt. Als Parameter wird der Name des Data Mart übergeben. Die Stored Procedure benutzt die momentane Datenbankverbindung, wobei die Datenbank mit Logging sein muss. Die im ersten Schritt gesammelten Probing Daten werden analysiert und daraus wird eine interne, relationale Darstellung einer Data Mart Definition erstellt, die dann in Tabellen in der aktuellen Datenbank abgespeichert wird. Diese Tabellen werden automatisch erzeugt, falls sie noch nicht existieren. Wird `ifx_probe2Mart()` ein weiteres Mal mit demselben Data Mart Namen aufgerufen, z.B. nachdem ein Probing neuer Abfragen stattgefunden hat, so wird die existierende Data Mart Definition in den Tabellen entsprechend den neuen Probing Daten ergänzt.

3. Erzeugen einer XML Datei aus einer internen, relationalen Data Mart Definition:

Dies erfolgt mit dem Aufruf der Stored Procedure `ifx_genMartDef()`. Solange die interne, relationale Darstellung der Data Mart Definition in den Tabellen der Logging-Datenbank existiert, kann dieser Schritt wiederholt werden. Allerdings wird eine schon existierende XML Datei überschrieben oder einen Fehler verursachen. Die Data Mart Definition in einer existierenden XML Datei wird nicht ergänzt. Die resultierende XML Datei enthält die minimale Definition des Data Mart, bestehend aus den Tabellen und deren Spalten, den Verknüpfungen der Tabellen, und der Information, welche Tabellen Faktentabellen sind.

4. Erzeugen eines Data Mart basierend auf der Definition in der XML Datei

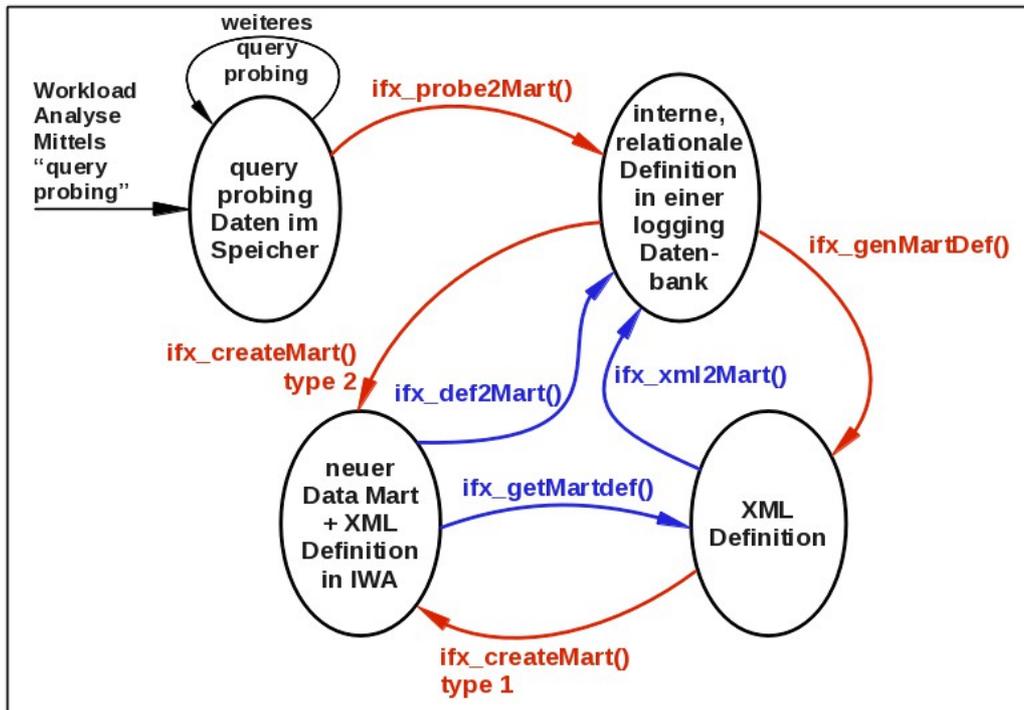
Schliesslich wird mit dem Aufruf der Stored Procedure `ifx_createMart()` Typ 1 der Data Mart im Accelerator erzeugt. Die Stored Procedure ergänzt die XML Definition um die Datentypen der einzelnen Spalten, etc., bevor sie dann zum Accelerator gesandt wird. IWA speichert die Data Mart Definition intern, wohingegen der Informix Server die XML Darstellung der Definition nicht speichert. Die XML Definition eines existierenden Data Marts kann vom Accelerator mit der Stored Procedure `ifx_getMartdef()` abgerufen werden.

Die Stored Procedure `ifx_createMart()` Typ 2 bietet eine Abkürzung an. Als Parameter dienen der Name des zu erzeugenden Data Mart und die Logging-Datenbank, in der zuvor die interne, relationale Darstellung der Data Mart Definition erzeugt wurde, z.B. mit `ifx_probe2Mart()`. Diese Abkürzung macht das Erzeugen der XML Datei überflüssig.

Die Stored Procedure `ifx_xml2Mart()` erwartet eine Datei mit einer XML Data Mart Definition als Eingabe und konvertiert diese XML Definition dann in eine interne, relationale Darstellung. Diese wird dann in den Tabellen der Logging-Datenbank abgespeichert. Existiert in den Tabellen schon eine Data Mart Definition zu diesem Data Mart, dann wird sie um die Elemente der XML Datei ergänzt.

Die Stored Procedure `ifx_def2Mart()` bietet eine weitere Abkürzung an, indem sie intern die XML Definition eines existierenden Data Mart direkt vom Accelerator abholt, diese in die interne, relationale Darstellung konvertiert und dann in den Tabellen der Logging-Datenbank abspeichert.

Die folgende Abbildung zeigt die verschiedenen Stadien einer Data Mart Definition und die Stored Procedures, mit denen sie zwischen den Stadien transportiert wird.



Stadien einer Data Mart Definition und zugehörige Stored Procedures

TechTipp: ENVIRONMENT – IFX_BAR_USE_DEDUP (nur mit TSM)

Die Umgebungsvariable IFX_BAR_USE_DEDUP ermöglicht die Nutzung der „deduplication capabilities“ des IBM Tivoli Storage Managers (TSM).

Die Variable ist nur wirksam, wenn sie als Umgebungsvariable beim Start der Informix Instanz gesetzt ist (unabhängig vom Wert, auf den sie gesetzt ist).

Der TSM unterstützt die Option der „deduplication“, um das Volumen der zu sichernden Objekte zu verringern. Hierbei eliminiert der TSM die Objekte der Sicherung, die in einer älteren Sicherung bereits vorhanden waren. Das Feature „deduplication“ muss im TSM explizit aktiviert werden. „Deduplication“ kann nicht in Zusammenspiel mit inkrementellen Sicherungen (Level 1 und Level 2) genutzt werden.

Hinweis: Sicherungen, die mit gesetzter Variable IFX_BAR_USE_DEDUP erstellt wurden, können nur „restored“ werden, wenn hierbei ebenfalls diese Variable gesetzt ist.

Der IBM Informix Primary Storage Manager (onpsm) unterstützt das Feature der „deduplication“ nicht.

TechTipp: Alter table InPlace für Serial8

Im Zeichen von BigData sind die Grenzen für Integerwerte schnell erreicht. Der Datentyp „Serial“ entspricht dem Datentyp „Integer“, der zusätzlich beim Einfügen eines neuen Datensatzes den nächst höheren Wert einer Zahlenfolge vergibt, falls das „Serialfeld“ nicht oder mit der Ziffer „null“ gefüllt wird. Ist das Limit von MAXINT (2'147'483'647) erreicht, so fängt die Zahlenvergabe wieder bei 1 an. Um dieses Limit zu umgehen, wird daher immer öfter auf den Datentyp „Serial8“ bzw. „BIGSERIAL“ übergegangen, der dem INT8 bzw. BIGINT entspricht.

Bisher wurde beim Ändern des Datentyp von Serial auf Serial8 die komplette Tabelle neu gespeichert, was zu erheblicher „Downtime“ durch die exklusive Sperre der Tabelle und zudem zu einem sehr hohen Logaufkommen geführt hat. Mit der Version 12.10.xC2 erfolgt die Umwandlung von Serial in Serial8 als „InPlace-Alter“ und somit mit einer minimalen Sperre der betroffenen Tabelle. Die reale Umwandlung des Datentyp in der Tabelle erfolgt nachgelagert, in Abhängigkeit welche Datenpages bei folgenden Transaktionen neu geschrieben werden müssen.

TechTipp: Pending „Alter Table“ ermitteln

Viele Aufrufe von „alter table“ können bereits als „InPlace-Alter“ durchgeführt werden, so dass ein „alter table“ nur zu einer sehr geringen Downtime führt, in der die Tabelle nicht für Änderungen zur Verfügung steht.

Um einen Überblick über die Tabellen zu erhalten, auf denen ein „alter table“ ausgeführt wurde, die aber noch nicht vollständig in der neuen Struktur gespeichert sind, kann eine Abfrage mittels SQL helfen.

Beispiel:

```
SELECT n.tabname as table,
       n.dbsname database,
       h.pta_oldvers as oldvers,
       h.pta_newvers as actvers,
       h.pta_totpgs as pend_pages
FROM sysptnhdr h, systabnames n
WHERE h.partnum = n.partnum
      AND h.pta_totpgs > 0
ORDER BY 5 desc
```

Ergebnis (im Beispiel):

table	person	
database	ibm	
oldvers	0	
actvers	3	← Die Tabelle wurde bereits 3x InPlace geändert
pendpages	1068	← 1068 Datenpages sind noch nicht in der aktuellsten Version gespeichert

TechTipp: LATERAL als Kennung für errechnete Tabellen

Informix unterstützt ab der aktuellen Version die Nutzung von LATERAL aus dem erweiterten ANSI-Standard. Errechnete Tabellen können als „lateral“ gekennzeichnet werden und sind somit bereits vor der Zeile der Erstellung im Select nutzbar.

Beispiel:

```
SELECT person.nam, person.tel, firma_name, firma_plz, firma_ort
FROM person,
     LATERAL (SELECT firma.name as firma_name,
                  firma.plz AS firma_plz,
                  firma.ort AS firma_ort
              FROM firma WHERE firma.fma = person.fma);
```

TechTipp: Unterstützung für .NET

Mit der aktuellen Version wurde die Unterstützung neuerer Versionen von .NET hinzugefügt, womit das Zusammenspiel mit ADO.NET verbessert wird.

Unterstützt werden aktuell die Versionen 2.0 bis 4.5 des .NET Framework.

Der Informix .NET Framework 2.0 Provider unterstützt .NET Framework 2.0, 3.0, und 3.5.

Der Informix .NET Framework 4.0 Provider unterstützt .NET Framework 4.0 und 4.5.

Der Support für .NET Framework 1.1 wurde eingestellt.

TechTipp: RSS und SDS um TimeSeries erweitert

Bisher war die Replikation von Instanzen, die Daten vom Typ TimeSeries enthalten, auf die HDR Replikation begrenzt. Mit der aktuellen Version wurde diese Einschränkung aufgehoben, so dass nunmehr auch RSS und SDS im Zusammenspiel mit TimeSeries genutzt werden können.

Die Enterprise Replication unterstützt die Replikation des Datentyps TimeSeries, wenn zuvor besondere Vorkehrungen getroffen wurden. Dieses Thema werden wir in der Ausgabe März des Informix Newsletter ausführlich mit Beispielen behandeln.

WebTipp: Smarter Cities – die Zukunft der Städte

Wer sich über die Möglichkeiten der Entwicklung von Städten informieren will, dem können wir die Seite „Smarter Cities“ empfehlen, auf der viele Aspekte aufgezeigt werden, bei denen IBM mit seinen Produkten die Lebensqualität steigern kann. INFORMIX ist ein wichtiger Bestandteil dieser Strategie.

Hier der Link, der wirklich lesenswert ist:

http://www.ibm.com/smarterplanet/de/de/smarter_cities/overview

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.iiug.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.bytec.de/de/software/ibm_software/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 IT-Specialist Informix Dynamic Server und DB2 UDB
IBM Software Group, Information Management
gerd.kaluzinski@de.ibm.com +49-175-228-1983

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

Sowie unterstützende Teams im Hintergrund.

Einige Anregungen für die Themen dieser Ausgabe wurden aus dem Informix Newsletter France übernommen.

Fotonachweis: Gerd Kaluzinski

(Redaktionsgarten)