Dokumentation Bibliometrische Indikatoren als gespeicherte Prozeduren

PL/SQL Package *indicator_queries* für die KB Bibliometriedatenbanken

Paul Donner, DZHW Berlin, 2016

Einführung

Der Hintergrund des Vorhabens war es, für gängige bibliometrische Indikatoren auf der Ebene von deutschen Institutionen aus der Bielefelder Institutionenkodierung - sowie für beliebig selbst definierte Publikationsmengen - Referenzimplementationen innerhalb der KB-Datenbankumgebung bereit zu stellen. Die hier beschriebenen Routinen können als Entwurf und Diskussionsgrundlage für eine künftige koordinierte Weiterentwicklung dienen. Damit könnte idealerweise die Benutzung der Datenbanken innerhalb des Benutzerkreises harmonisiert werden. Davon abgesehen können gespeicherte Prozeduren die Benutzung der Datenbank erleichtern und beschleunigen und Fehler verhindern.

Weiterhin soll Benutzern über eine allgemeine Beschreibung der Programmierlogik der vorliegenden Funktionen das Erstellen von eigenen, spezielleren Funktionen nach dem gleichen Grundmuster erleichtert werden.

Die Funktionen sind programmiert in der Sprache PL/SQL, der prozeduralen Erweiterung von SQL von Oracle, welche komplett in das Datenbanksystem integriert ist. Die Indikatorfunktionen sind als ein PL/SQL *Package* zusammengefasst. Diese entsprechen libraries in anderen Programmiersprachen. Packages bestehen aus den beiden Teilen *Specification*, mit der Definition von Funktionen, Variablentypen und Variablen, und *Body*, mit den Implementationen der Funktionen. Entsprechend hat das *indicator_queries* Package eine Specification- und eine Body-Datei und daneben eine Datei mit Anwendungsbeispielen.

Installation

Die Quellcodedateien können Sie von Herrn Donner beziehen oder aus dem gitlab Versionierungsrepository des Projektes (https://gitlab.com/pdonner/kb_indicator_procedures/). Wenn Sie über ein gitlab-Benutzerkonto verfügen, können Sie von dort die Dateien kopieren, wenn Herr Donner Sie zu dem Projekt als Beteiligten hinzufügt, da es nicht öffentlich zugänglich ist. Dort können Sie auch Issues für Bugs und Änderungswünsche anlegen. Wenn Sie wünschen, können Sie über das gitlab-Projekt auch konkrete Änderungen in Form von Programmcode beisteuern.

Zur Installation des Paketes unter Ihrem Benutzerschema kopieren Sie erst den Code der Specification-Datei in ein mit der Datenbank verbundenes Worksheet des SQL Developer bzw. SQL*Plus und führen den Code dort aus, danach tun Sie das gleiche für die Body-Datei. Um das Package überhaupt verwenden zu können, muss Ihr Benutzeraccount user level Zugriffsrechte auf die im Package referenzierten Tabellen der Datenbank haben. Momentan ist die referenzierte

Bibiometriedatenbank-Version *wos12b*. Standardmäßig haben Benutzeraccounts in den KB Datenbanken group level Zugriffsrechte auf die KB-BDB- und Roh-DB-Schemas. Deswegen müssen Sie die Kollegen vom FIZ um diese Zugriffsrechte auf die Tabellen des entsprechenden Schemas bitten, was kein Problem darstellt. Sie können die benutzte BDB-Version des Packages ändern, indem Sie alle Referenzen auf wos12b ersetzen. Dann brauchen Sie auch für jenes Schema user level Zugriffsprivilegien.

Benutzung

Die Indikatorfunktionen sind sogenannte *pipelined table functions*, was bedeutet, dass ihre Ausgabe wie eine Tabelle in der FROM clause eines SQL statements benutzt werden kann. Ein einfaches Beispiel ist dieses:

```
SELECT *
FROM TABLE(indicator_queries.inst_itemcounts(105, 2000, 2010)) j;
```

Diese Funktion nimmt drei Parameter entgegen und gibt als Ausgabe eine virtuelle Tabelle mit drei Spalten zurück. Parameter und Rückgabewert sind in der Deklaration der Funktion in der Package Specification festgelegt. Der Typ des Rückgabewerts ist jeweils funktionsspezifisch als PL/SQL Variablentype Collection mit bestimmter Struktur definiert. Die Ausgabetabelle kann wie eine normale Tabelle benutzt werden, z. B. können die Spalten in der SELECT clause eingeschränkt oder weiterverrechnet werden, Zeilen in der WHERE clause gefiltert werden und JOINS durchgeführt werden.

Im obigen Beispiel sind die Spaltennamen PUBYEAR, ITEMS_WHOLE und ITEMS_FRACT. Da hinter den TABLE()-Aufruf den Namensalias „j“ gesetzt ist, kann die erste Spalte wie folgt in einem JOIN benutzt werden:

```
SELECT j.pubyear, j.items_whole, k.items_whole
FROM TABLE(indicator_queries.inst_itemcounts(105, 2000, 2010)) j
JOIN TABLE(indicator_queries.inst_itemcounts(129, 2000, 2010)) k
ON j.pubyear = k.pubyear;
```

Die Parameter können zur besseren Verständlichkeit auch mit ihren Namen übergeben werden:

```
SELECT * FROM
TABLE(indicator_queries.inst_itemcounts(inst => 105, py_start => 2000,
py_end => 2010)) j;
```

Generell ist es aber ausreichend, die Parameter in der korrekten Reihenfolge zu übergeben. Für bestimmte Parameter gibt es auch voreingestellte Standardwerte, die benutzt werden, wenn kein Parameterwert übergeben wird. Funktionen, die `inst_*` heißen, sind gedacht für Institutionen aus der Bielefelder Institutionenkodierung. Ihr erster Parameter ist immer die ID (pk_kb_inst) der Institution. Funktionen, die `set_*` heißen, können für beliebige Publikationssets benutzt werden. Sie haben einen einzigen Eingabeparameter, einen benutzerdefinierten Cursor mit dem gewünschten Set an Item-Identifiern (pk_items), z. B.:

```
SELECT * FROM TABLE(indicator_queries.set_itemcounts(
  CURSOR(SELECT DISTINCT fk_items FROM wos12b.kb_s_wos_addr_inst
    WHERE fk_kb_inst = 1024)
));
```

Für viele der `inst_*`-Funktionen gibt es analoge `set_*`-Funktionen, nämlich immer dann, wenn eine solche auch sinnvoll ist. Es gibt außerdem eine Hilfsfunktion um die Ergebnismenge für die `set_*`-Funktionen zwischen zu speichern, damit nicht mehrfach nacheinander für Indikatorfunktionen der gleiche Cursor übergeben und neu ausgeführt werden muss. Zur Benutzung muss die Funktion `define_pubset()` einmal zu Beginn einer Session ausgeführt werden. Anschließend kann innerhalb der Session auf die `set_*`-Funktionen ohne Parameter zugegriffen werden:

```
SELECT indicator_queries.define_pubset(CURSOR(SELECT DISTINCT fk_items FROM
wos12b.kb_s_wos_addr_inst WHERE fk_kb_inst = 129)) publications_set_defined
FROM DUAL;
SELECT * FROM TABLE(indicator_queries.set_itemcounts);
```

Besonderheit bei klassifikationsbasierten Indikatoren

Es gibt in WoS (wos12b) eine geringe Anzahl von Items, die keine Zuordnung zu einer Subject Category haben. Dies führt dazu, dass diese bei der Berechnung von feldnormalisierten Indikatoren nicht berücksichtigt werden können. Die Indikatorenfunktionen, bei denen dies eine Rolle spielt, haben deswegen in ihrer Rückgabetabelle die zusätzliche Spalte included_publications, in der genau angegeben ist, wie viele Publikationen in die Berechnung eingegangen sind. Zum Vergleich sollte daher die Rückgabetabelle der relevanten *_itemscounts-Funktion mit einem Join hinzugezogen werden.

Package-Einstellungsvariablen

Für die üblichen Merkmale, die zur bibliometrischen Abgrenzung von Publikationsmengen benutzt werden, gibt es in der Package-Specification vordefinierte Variablen, die vom Benutzer an den jeweiligen Anwendungsfall angepasst werden können:

Variable	Zweck
current_year	Wird benutzt, um das maximal berechenbare Jahr bei Zitationsindikatoren zu berechnen.
default_py_start	Beginn des untersuchten Publikationszeitraums
default_py_end	Ende des untersuchten Publikationszeitraums
doctypes_list	Berücksichtigte Dokumenttypen
pubtypes_list	Berücksichtigte Publikationstypen
datasource_list	Berücksichtigte Data Source (WoS)
hc_threshold	Perzentil für highly cited rate, z.B. 0.05 für die Benutzung der 5 % am häufigsten zitierten Publikationen als Kriterium

Es ist zu beachten, dass die Definition dieser Variablen auf Package-Ebene den Nachteil hat, dass sie nicht ad hoc geändert werden können, sondern fixiert sind, und eine Änderung der Variablenwerte eine Neukompilierung des Packages nach sich zieht.

Allgemeines Muster der Programmierlogik

Alle Indikatorfunktionen folgen dem gleichen Aufbau, der leicht übernommen und für weitere Funktionen adaptiert werden kann. Jede Funktion braucht eine Collectionvariable eines

vordefinierten Typs, in die Ergebnisse einer SQL-Query zwischengespeichert werden. Diese ist immer zwischen IS und BEGIN deklariert. Der Variablentyp ist in der Specification definiert. In jeder Funktion gibt es eine zentrale SQL-Query, die die Daten berechnet und das Ergebnis in die Collectionvariable mit dem Befehl BULK COLLECT INTO var … übergibt. Am Ende jeder Funktion wird das Ergebnis aus der Variable zeilenweise ausgegeben über die Konstruktion:

```
    FOR i IN 1 .. var.COUNT LOOP
      PIPE ROW(var(i));
    END LOOP;
  RETURN;
```

Skalare Parameter, z. B. Anfangsjahr, Endjahr, Land, Institutionen-ID, werden als skalare Variablen übergeben und in der Query direkt referenziert. IN-Listen, wie z. B. die Auflistung gewünschter Dokumenttypen, werden über die Konstruktion wie diese verwendet:

```
WHERE doctype IN (SELECT * FROM TABLE(doctypes_list))
```

Die Berechnungslogik von Indikatoren kann über die Beschränkungen von SQL hinausgehen, indem SQL Queries benutzt werden um die Rohdaten zu wählen und die eigentliche Berechnung über PL/SQL gemacht wird.

Funktionsreferenz

Funktion/Funktionsfamilie	**verfügbar für:**			**Beschreibung**
	inst	**set**	**country**	
define_pubset	trifft nicht zu			Festlegung einer Publikationsmenge für die Sitzung
*_itemcounts	+	+	+	Anzahl Artikel, full und fractional count, wobei fractional count Berechnung davon abhängt, ob Institution oder Land
*_unc_rate	+	+	+	Anteil unzitierter Artikel
*_mfncr	+	+	+	Mittlere feldnormalisierte Zitationsrate
*_cpp	+	+	-	mittlere Zitationsrate
*_hcr	+	+	+	Anteil hochzitierter Publikationen (highly cited rate)
inst_ext_coop	+	-	-	Anteil Publikationen mit deutscher Kooperation
*_int_coop	+	+	+	Anteil Publikationen mit internationaler Kooperation
inst_strnat_coop	+	-	-	Anteil Publikationen mit deutscher Kooperation aber ohne internationale Kooperation („strictly national“)
inst_sect_coop	+	-	-	Anzahl u. Anteil Kooperationsbeziehungen nach Sektoren (nicht Publikationen)
inst_inst_sect_coop	+	-	-	Anzahl kooperierender deutscher Einrichtungen nach Sektoren
*_country_coop	+	-	+	Anzahl Ko-Publikationen mit anderen Ländern