projekt kivitendo - Unterstützung #108

Ungenutzte Indexe ausmisten

19.11.2015 13:49 - G. Richardson

Status:	Neu	**Beginn:**	19.11.2015
Priorität:	Niedrig	**Abgabedatum:**	
Zugewiesen an:		**% erledigt:**	0%
Kategorie:		**Geschätzter Aufwand:**	0.00 Stunde
Zielversion:		**Aufgewendete Zeit:**	0.00 Stunde

Beschreibung

In 34ea9b39f4 habe ich schon ein paar doppelte Indexe rausgeschmissen, weil es in den Tabellen mittlerweile einen Primary Key gibt, und die händisch gesetzten deshalb nicht mehr nötig waren. Es gibt aber sicherlich noch mehr, was man aufräumen könnte. Z.B. gibt es einen Index auf lower(source) in acc_trans, der nur greift, wenn man in einer Abfrage nach einem eindeutigen source sucht, z.B.:

```
select * from acc_trans where lower(source) = 'Rechnung 55506505';
```

Sobald man in der Abfrage das lower wegläßt, oder ein like benutzt, ist der Index schon nutzlos. Ich konnte keine aktuelle Codestelle finden, wo das so benutzt wird, das stammt vermutlich also noch aus alten Zeiten. In den Datenbanken, wo ich nachgeschaut habe, ist das sogar der größte Index, bei einem Mandanten knapp 3MB groß, und wurde noch nie benutzt. Ich würde den Index also gerne entfernen, und nehme gerne Vorschläge entgegen, was noch entfernt werden könnte/sollte.

Hier kann man sich die aktuell unbenutzten Indexe und deren Größe anschauen:

```
SELECT
    relid::regclass AS table,
    indexrelid::regclass AS index,
    pg_size_pretty(pg_relation_size(indexrelid::regclass)) AS index_size,
    idx_tup_read,
    idx_tup_fetch,
    idx_scan
FROM
    pg_stat_user_indexes
    JOIN pg_index USING (indexrelid)
WHERE
    idx_scan = 0
    AND indisunique IS FALSE;
```

Historie

#1 - 19.11.2015 15:37 - G. Richardson

Nochmal zu dem lower(g.reference), das wird schon irgendwie als lower verwendet, und zwar beim Buchungsjournal in GL/all_transactions, wenn man beim Buchungsjournal im Referenzfeld etwas eingibt. Dort wird dann allerdings das lower(reference) ausgelesen, aber nicht verwendet, stattdessen macht der Filter:

```
  if ($form->{reference}) {
    $glwhere .= qq| AND g.reference ILIKE ?|;
    $arwhere .= qq| AND a.invnumber ILIKE ?|;
    $apwhere .= qq| AND a.invnumber ILIKE ?|;
    push(@glvalues, '%' . $form->{reference} . '%');
    push(@arvalues, '%' . $form->{reference} . '%');
    push(@apvalues, '%' . $form->{reference} . '%');
  }
```

Damit der Index verwendet werden kann müßte man folgendes machen:

```
  if ($form->{reference}) {
    $glwhere .= qq| AND lower(g.reference) = ?|;
    $arwhere .= qq| AND lower(a.invnumber) = ?|;
    $apwhere .= qq| AND lower(a.invnumber) = ?|;
    push(@glvalues, $form->{reference} );
```

```
    push(@arvalues, $form->{reference} );
    push(@apvalues, $form->{reference} );
  }
```

Dann wäre die Abfrage wahrscheinlich etwas schneller, aber man könnte im Buchungsjournal nicht mehr nach Teilstrings suchen, was die Nützlichkeit für den Anwender stark einschränken würde.

Das Gleiche trifft vermutlich für die Indexe gl_reference_key, customer_name_key und vendor_name_key, wo man im Feld Beschreibung die Beschreibung der Dialogbuchung oder den Kunden- oder Lieferantennamen eingeben kann. Statt

```
 if ($form->{description}) {
   $glwhere .= " AND g.description ILIKE ?";
   $arwhere .= " AND ct.name ILIKE ?";
   $apwhere .= " AND ct.name ILIKE ?";
   push(@glvalues, '%' . $form->{description} . '%');
   push(@arvalues, '%' . $form->{description} . '%');
   push(@apvalues, '%' . $form->{description} . '%');
 }
```

wäre also Folgendes nötig, damit der Index bei der Suche hilft:

```
  if ($form->{description}) {
    $glwhere .= " AND lower(g.description) = ?";
    $arwhere .= " AND ct.name = ?";
    $apwhere .= " AND ct.name = ?";
    push(@glvalues, $form->{description} );
    push(@arvalues, $form->{description} );
    push(@apvalues, $form->{description} );
  }
```

Von der Funktionialität her würde ich das ILIKE aber auf jeden Fall bevorzugen, und die unnützen Indexe entfernen (wobei ich hier noch nicht geprüft habe, ob die nicht doch vielleicht woanders verwendet werden).