

projekt kivitendo - Fehler #126

bankbuchung rückgängig machen

06.02.2016 11:56 - Jan Büren

Status:	Neu	Beginn:	06.02.2016
Priorität:	Normal	Abgabedatum:	
Zugewiesen an:		% erledigt:	0%
Kategorie:		Geschätzter Aufwand:	0:00 Stunde
Zielversion:		Aufgewendete Zeit:	0:00 Stunde

Beschreibung

Eine Zahlung ist falsch zugeordnet.

Im Schritt 1 hab ich die Verknüpfung im Kontenabgleich aufgehoben.

i.O. -> bank_transactions.cleared = false

Im Schritt 2 möchte ich die Zahlung neu zuordnen.

Ich hab die verknüpfte Zahlung gelöscht (über verknüpfte Belege)

Da ist die erste Unstimmigkeit, die Verknüpfung ist weg, aber die Zahlung noch vorhanden.

Das ist an der Stelle nicht konsistent.

Ferner sehe ich die Banktransaktion nicht in der Auswahlliste der zu noch verbuchenden Transaktion, obwohl die Zuordnung wieder aufgehoben ist.

Sprich die Zahlung scheint verschwunden zu sein - Und der Mitarbeiter wird unsicher, weil einmal falsch verknüpfte Zahlungen nicht mehr reparierbar sind.

Wahrscheinlich ist diese Stelle in BankTransaction hierfür verantwortlich:

```
(where => [ amount => {ne => \'invoice_amount'},
```

Dann hab ich also drei verschiedene Stellen, die den Zustand bestimmen:

- a) reconciliation_links
- b) invoice_amount in bank_transactions
- c) cleared in bank_transactions

Feld b) überhaupt zu verwenden, finde ich genauso intelligent wie paid in ar / ap.

a) und c) sind schon ausreichend, um den Status des Zahlungszustands idiotensicher zu bestimmen.

Ich kann hiermit auch schon alle Fälle abdecken:

- 1) Eine Zahlung geht auf eine Rechnung
- 2) Eine Zahlung geht auf mehrere Rechnungen
- 3) Eine Rechnung hat mehrere Zahlungen

Ich weiß hierüber auch schon exakt welche Geldmenge welcher Rechnung zugewiesen ist, mit nur dem gesamt zugewiesenen invoice_amount geht das auf dem ersten Blick nicht.

Mein Vorschlag wäre statt dieser Abfrage:

```
(where => [ amount => {ne => \'invoice_amount'},
```

die in etwa so (pseudocode) zu ersetzen:

```
(where => [ amount => { ne => sum (amount) from acc_trans where acc_trans_id in (select acc_trans_id from reconciliation_links rl where rl.bank_transaction_id = bank_transaction.id }
```

Vom Datenmodell her, wäre es sinnvoll wenn reconciliation_links nicht einfach gelöscht werden könnten, solange die noch auf acc_trans_ids verweisen.

Damit wäre mein Fehler an der Oberfläche zumindestens erstmal hart ausgeschlossen gewesen.

Die aktuelle Verknüpfung ist wie folgt:

```
acc_trans <- reconciliation_links -> bank_transaction
```

D.h. der Link ist der Vater von beiden Einträgen. Aktuell wird der Link aber nicht immer gesetzt (Kontenabgleich im Dialog) und ist ferner eher die schwächste Stelle in dieser Kette, d.h. sie kann simpel über "Verknüpfung entfernen" mittels dem verknüpften Belege-Dialog entfernt werden.

Sinnvollere Väter oder bessere Hühner vor den Eiern ...

Bspw.:

```
bank_transaction -> reconciliation_links -> acc_trans
```

Ich kann die Zahlung nicht löschen, bevor ich den Verweis rausgenommen habe, ferner kann ich den Verweis nicht rausnehmen, bevor ich die Transaktion nicht anfasse.

Die Änderung kann in der Tat über Kontenabgleich rückgängig machen in einem Schritt passieren. Am Besten noch mit einem entsprechenden Dialog.

Historie

#1 - 07.02.2016 18:43 - Jan Büren

ich hab jetzt noch eine Blick tiefer geschaut ...

wenn man die referentielle integrität jetzt ignoriert, die aktuell sowieso nicht sinnvoll ist - kann man die tabelle reconiciliation_links auch wegwerfen und die, genau hierfür vorgesehenen Tabelle record_links nutzen.

... vielleicht ergänzt jmd. ideen!

danke

#2 - 24.02.2016 11:38 - Jan Büren

Jetzt erstmal ein fettes Lob. Ich hab zum ersten Mal den Import für unsere kivi selber genutzt und der Import filtert Duplikate raus und die Vorschlagsliste lässt sich wirklich schnell wegklicken und verarbeiten.

Aufgefallen ist mir allerdings noch:

a) Keine Unterscheidung zwischen Zahlungsausgang oder Zahlungseingang

Das ist insofern unglücklich für Mandanten die sowohl eine Kunden als auch eine Lieferantenbeziehung zu derselben Firma pflegen. Hier könnte die Zuordnung schon so schlau sein, dass Zahlungsausgänge eher nicht zu Verkaufsrechnungen passen können ...

b) Zahlungen umbuchen

s.o. Aktuell können Korrekturen nur von kivi-Administratoren auf DB-Ebene vorgenommen werden.

#3 - 23.05.2016 15:20 - Jan Büren

Zahlungen löschen ist etwas schwierig, super wäre es hierfür auch einen Helfer zu haben.

```
SL::DB::Helper::Payments->undo_payments
```

oder sowas.

Das schwierige ist allerdings die Gegenbuchung zu finden, hat jemand hier eine bessere idee als nach transdate,gldate,itime zu prüfen?

```
10323 | 1655 | 17 | 19.04000 | 2016-04-29 | 2016-05-10 | | f | f
| f | | f | | | 0 | 2016-05-10 12:34:45.660568 |
| AR | | 0
10322 | 1655 | 30109 | -19.04000 | 2016-04-29 | 2016-05-10 | | t | f
| f | | f | | | 0 | 2016-05-10 12:34:45.660568 | 2016-05-10 12:47
:12.148686 | AR_paid:AP_paid
```

Vielleicht kann man prinzipiell die die Paare besser in acc_trans kennzeichnen?

Ansonsten hab ich nur sowas zur Auswahl:

```
my $id = SL::DB::Manager::AccTransaction->find_by(trans_id => 1655, chart_link => 'AR', amount => 19.04);
Wobei amount = acc_trans.amount (von_paid) * -1
```

Und für dein Einkauf genau spiegelverkehrt.

#4 - 24.05.2016 17:20 - Jan Büren

Ich hab mir das noch weiter überlegt.

Es hilft alles leider nicht wirklich, die Verknüpfung von Bankbuchungen nach AccTrans muss einfach relational sicher vorhanden sein.

Ich hab in bank_transactions einfach Daten die zugeordnet sind: invoice_amount = amount und die ich aber über RecordLinks nicht finden kann (ach ...)

In diesem Fall sind die mit GL verknüpft, aber auch hier spricht überhaupt nichts dagegen, diese Buchungen mit bank_transactions zu verbinden und nicht

nur über den Kontenabgleich.

Die Zahlungskonten bekommen genauso einen chart_link in acc_trans zugewiesen.

Ach mann, das wirft eine ganze Reihe an tieferen Problemchen auf:

a) record_links sind nett, aber ungeeignet um Bankbewegung sicher mit Buchungsbewegungen zu verbinden

ich würde mir hier lieber eine neue Verknüpfungstabelle wünschen, die auch von record_links angezeigt werden kann

b) warum ist gl nicht mit bank_transactions verknüpft?

Das ist schon vom Design doof, hier nicht den gleichen Mechanismus wie bei ar/ap zu verwenden.

Ansonsten erstmal mein Konsolen-Skript um Bank-Verbuchungen für die Verkaufsseite wieder rückgängig zu machen, als erste Ideen-Skizze:

```
use SL::DB::Helper::LinkedRecords;
use SL::DB::Manager::BankTransaction;
use SL::DB::RecordLink;
use SL::DB::Manager::AccTransaction;
# get wrong transaction by purpose
my $bank_transactions = SL::DB::Manager::BankTransaction->find_by(purpose => 'KNR 20003559 RENR 9106 19 RENR 9
10620');

# get all linked ids an delete paid

#my @linked_objects= $bank_transactions->linked_records->to_id;

# delete all payments with bank_id

#foreach my $linked_bank_transaction (@linked_objects) {
#  $linked_bank_transaction->id;
#}

my @to_id = SL::DB::Manager::RecordLink->get_all(where => [
  from_table => 'bank_transactions',
  from_id => $bank_transactions->id,
]);
#foreach my $rl (@to_id) {
#  foreach my $id (@$rl) {
#    $id->to_id;
#  }
#}
my (@to_ids, @acc_trans_payment);
push (@to_ids, $_->to_id for @{$to_id[0]});
push (@to_ids, 1646);

foreach my $id (@to_ids) {
  # get all
  # http://redmine.kivitendo-premium.de/issues/126
  my $current_payment = SL::DB::Manager::AccTransaction->find_by(trans_id => $id, chart_link => {ILIKE => '%_p
aid%'});
  # check for scalar
  my $ml = $current_payment->get_type eq 'ar' ? -1 : 1;
  my $current_ARAP = SL::DB::Manager::AccTransaction->find_by(trans_id => $id, chart_link => 'AR', amount =
> $current_payment->amount * $ml);
  # !!!! no more than 1 !!!! check for scalar
  # SL::DB::Manager::AccTransaction->delete_all(where => [ trans_id => $id, chart_link => {ILIKE => '%_paid%'}]
);
  # set ar ap paid
  my $inv = SL::DB::Manager::Invoice->find_by(id => $id);
  $inv->paid($inv->paid + $current_payment->amount);
  $inv->save;
  $current_payment->delete;
  $current_ARAP->delete;
}
# SL::DB::Manager::AccTransaction->delete_all(where => [ trans_id => $_, chart_link => {ILIKE => '%_paid%'}])
foreach (@to_ids);

# unlink all
```

```
SL::DB::Manager::RecordLink->delete_all(where => [ from_table => 'bank_transactions', from_id => $bank_transactions->id]);
```

```
$bank_transactions->invoice_amount(0);  
$bank_transactions->save;
```

#5 - 25.05.2016 11:09 - Jan Büren

Die nächste Ergänzung zu diesem Punkt, super wäre es, anstatt der fragilen RecordLinks eine andere Verknüpfungstabelle zu haben:

```
# Record a record link from the bank transaction to the invoice  
my @props = (  
  from_table => 'bank_transactions',  
  from_id    => $bt_id,  
  to_table   => $invoice->is_sales ? 'ar' : 'ap',  
  to_id      => $invoice->id,  
);
```

```
SL::DB::RecordLink->new(@props)->save;  
}  
$bank_transaction->save;  
}
```

```
# Record a record link from the bank transaction to the invoice  
my @props = (  
  from_id    => $bt_id,  
  to_id      => $acc_trans_id,  
);
```

```
SL::DB::BankTransactionLink->new(@props)->save;  
}  
$bank_transaction->save;  
}
```

Inhaltlich müsste die Tabelle dann ähnlich wie die ReconciliationLinks aufgebaut sein.

Frage, kann man diese Verknüpfung an der Oberfläche (einfach) als alternativen Link anzeigen lassen?