

Régulariser les transactions cynod qui n'ont pas généré des écritures comptables

Suite à un incident cynod, il peut arriver que des transactions validées (statut **VA**) durant le processus ne génèrent pas des écritures comptables. L'impact est majeur dans le sens où le prestataire fourni un service pour lequel le bénéficiaire n'est pas réellement débité du montant de la transaction alors que le solde est impacté.

Les cas identifiés à ce jour concernent les transactions :

- carte en online – **fixed**
- carte en offline puis synchronisée
- wallet –**fixed**
- ticket en offline

Ce guide a pour objet de définir les différentes étapes d'identification et de correction des transactions n'ayant pas généré des écritures. Seules les transactions ayant été validées sont concernées. Les transactions d'annulation (**AN**) ou celles nécessitant une validation (**SA**) devront faire l'objet d'un traitement spécifique soit par le client soit par l'agent support.

Avertissement : vous allez réaliser des actions sensibles. Il est important de s'assurer que toutes les actions décrites soient scrupuleusement respectées.

Étape 1 : Extraire la situation avant traitement

1. Lancer le report **checkEcrituresTrx** depuis [jasperserver cynod](#) pour la base du client concerné
2. Définir la période voulue

Il est aussi possible d'exécuter la requête SQL suivante

[checkEcrituresTrx.sql](#)

```
SELECT a.wallet_id, a.id, a.carte_emettrice_id, a.carte_receptrice_id,
a.code_evenement,a.code_utilisateur_saisie,a.date_enregistrement_transaction,a.date_transaction,
a.ind_annulation,a.is_online,a.montant_transaction,a.motif,a.motif_annulation,a.numero_operation,
a.operation_offline_id,a.produit,a.ref_terminal,a.report_cumul_consommation_offline,
a.report_cumul_consommation_synchronise,a.report_plafond_offline,a.report_solde_online_on_card,
a.solde_apres_emmission,a.solde_apres_reception,a.solde_avant_emmission,
a.solde_avant_reception,
a.statut,a.uid_card,a.ticket_carburant_id,a.version_mobile,a.volume,
a.statut_remontee_annulation,a.statut_synchro_to_etab,
a.ind_espece,a.ind_trx_forcage_inter_oper,a.deleted+'0',
b.num_carte AS numCarteEmetrice ,b.intitule AS
intituleCarteEmetrice,b.statut AS 'statut_carte_emetrice',
```

```
c.num_carte AS numCarteReceptrice ,c.intitule AS  
intituleCarteReceptrice,c.statut AS 'statut_carte_receptrice'  
FROM (operation a LEFT OUTER JOIN carte b ON a.carte_emettrice_id=b.id)  
LEFT OUTER JOIN carte c ON a.carte_receptrice_id=c.id  
WHERE a.id NOT IN (SELECT b.operation_id FROM ecriture b)  
AND a.date_transaction >= '2021-01-01'  
ORDER BY a.date_transaction ;
```

Étape 2 : Préparer les données à régulariser

Exécuter les requêtes SQL suivantes sur la base de données du client

[scriptRegulePreTraitement.sql](#)

```
-- -----  
-- Insertion des opérations de régul transactions cartes  
-- -----  
INSERT INTO operation_regul(  
version,statut,montant_transaction,operation_id,TYPE,date_transaction,n  
umero_operation)  
SELECT  
0,'SA',a.montant_transaction,a.id,'C',a.date_transaction,a.numero_opera  
tion  
FROM operation a LEFT OUTER JOIN carte b ON a.carte_emettrice_id=b.id  
WHERE a.id NOT IN (SELECT b.operation_id FROM ecriture b)  
AND a.date_transaction >= '2021-01-01'  
AND a.ticket_carburant_id IS NULL  
AND a.id NOT IN (SELECT operation_id FROM operation_regul)  
AND a.statut = 'VA'  
ORDER BY a.date_transaction ;  
  
-- -----  
-- Insertion des opérations de régul transactions tickets  
-- -----  
INSERT INTO operation_regul(  
version,statut,montant_transaction,operation_id,TYPE,date_transaction,n  
umero_operation)  
SELECT  
0,'SA',a.montant_transaction,a.id,'T',a.date_transaction,a.numero_opera  
tion  
FROM operation a LEFT OUTER JOIN carte b ON a.carte_emettrice_id=b.id  
WHERE a.id NOT IN (SELECT b.operation_id FROM ecriture b)  
AND a.date_transaction >= '2021-01-01'  
AND a.ticket_carburant_id IS NOT NULL  
AND a.id NOT IN (SELECT operation_id FROM operation_regul)  
AND a.statut = 'VA'
```

```
ORDER BY a.date_transaction ;
```

Étape 3 : Rendre les cartes à régulariser débitrices

[scriptRegulePreTraitement_2.sql](#)

```
-- -----  
-- Sauvegarde du statut débitrice de la carte pour forcer les débits  
-- -----  
UPDATE operation_regul a, operation b, carte c , carte d  
SET  
    a.is_carte_emetteur_debitrice=c.is_debitrice,  
    a.is_carte_recepteur_debitrice=d.is_debitrice  
WHERE a.operation_id=b.id  
AND   b.carte_emettrice_id=c.id  
AND   b.carte_receptrice_id=d.id  
AND   a.type='C';  
  
UPDATE carte a, operation b, operation_regul c  
SET  
    a.is_debitrice = TRUE  
WHERE c.operation_id=b.id  
AND   b.carte_emettrice_id=a.id  
AND   c.type='C';  
  
-- -----  
-- Forcer les cartes à être débitrices  
-- -----  
UPDATE carte a, operation b, operation_regul c  
SET  
    a.is_debitrice = TRUE  
WHERE c.operation_id=b.id  
AND   b.carte_receptrice_id=a.id  
AND   c.type='C';  
  
UPDATE carte a, operation b, operation_regul c  
SET  
    a.is_debitrice = TRUE  
WHERE c.operation_id=b.id  
AND   b.carte_emettrice_id=a.id  
AND   c.type='C';
```

Étape 4 : Exécuter le programme de régularisation

Pre requis

1. Avoir un compte d'accès à la plateforme web
2. Disposer du rôle **ROLE_CHECK_CARTE**
3. Etape 2, 3 obligatoires

Pour les cartes

1. Se connecter à la plateforme
2. Se rendre sur la page **Check information** en saisissant au niveau de l'URL de la barre du navigateur /check
3. Cliquer sur **Regule écriture (Carte et ticket)**
4. Cliquer sur **Carte**

Le programme va s'exécuter. Il faudra patienter. Si cela ne passe pas, directement saisir `check/regulEcriture?typeService=C`

Pour les tickets

1. Se connecter à la plateforme
2. Se rendre sur la page **Check information** en saisissant au niveau de l'URL de la barre du navigateur /check
3. Cliquer sur **Regule écriture (Carte et ticket)**
4. Cliquer sur **Ticket**

Le programme va s'exécuter. Il faudra patienter. Si cela ne passe pas, directement saisir `check/regulEcriture?typeService=T`

Étape 5 : Extraire les statistiques de traitement

[scriptRegulePostTraitement.sql](#)

```
-- -----  
-- statistiques traitement programme regul  
-- -----  
SELECT TYPE, statut, COUNT(*)  
FROM operation_regul  
GROUP BY TYPE, statut;  
  
-- -----  
-- Motif non ecriture pendant regul carte  
-- -----  
SELECT remarque, COUNT(*)  
FROM operation_regul  
WHERE statut='RJ'
```

```
AND TYPE='C'
GROUP BY remarque;

-- -----
-- Motif non ecriture pendant regul ticket
-- -----

SELECT remarque, COUNT(*)
FROM operation_regul
WHERE statut='RJ'
AND TYPE='T'
GROUP BY remarque;
```

Étape 6 : Remettre les cartes débitrices à leur état initial

[scriptRegulePostTraitement_2.sql](#)

```
-- -----
-- Rollback forçage des cartes à être débitrices
-- -----

UPDATE carte a, operation b, operation_regul c
SET
    a.is_debitrice = c.is_carte_emetteur_debitrice
WHERE c.operation_id=b.id
AND    b.carte_emettrice_id=a.id
AND    c.type='C';

UPDATE carte a, operation b, operation_regul c
SET
    a.is_debitrice = c.is_carte_recepteur_debitrice
WHERE c.operation_id=b.id
AND    b.carte_receptrice_id=a.id
AND    c.type='C';
```

Étape 7 : Mise à jour des soldes des cartes sur la base de la somme des écritures

[scriptRegulePostTraitement_3.sql](#)

```
-- -----
-- Calcul du solde basé sur la somme des écritures
-- -----

DROP TABLE IF EXISTS tmp_maj_solde_carte;
CREATE TABLE tmp_maj_solde_carte
AS SELECT carte_id,SUM(valeur_ecriture) AS solde_ecriture
FROM ecriture
```

```
GROUP BY carte_id;

ALTER TABLE tmp_maj_solde_carte
    ADD INDEX idx_tmp_maj_solde_carte (carte_id ASC);

-- -----
-- Sauvegarde des soldes erronées avant mise à jour
-- -----

DROP TABLE IF EXISTS tmp_sav_ecart_solde ;
CREATE TABLE tmp_sav_ecart_solde
AS
SELECT
a.num_carte,a.intitule,a.cumul_consommation_offline,a.cumul_consommatio
n_synchronise,a.solde_online, a.deleted,a.statut,
a.ind_annulation,a.type_carte,
b.solde_ecriture
FROM carte a, tmp_maj_solde_carte b
WHERE a.id=b.carte_id
AND a.solde_online <> b.solde_ecriture;

-- -----
-- Mise à jour des soldes
-- -----

UPDATE carte a, tmp_maj_solde_carte b
SET a.solde_online = b.solde_ecriture
WHERE a.id=b.carte_id
AND a.solde_online <> b.solde_ecriture;
```

Étape 8 : Extraire la situation après traitement

1. Lancer le report **checkEcrituresTrx** depuis [jasperserver cynod](#) pour la base du client concerné
2. Définir la période voulue

Il est aussi possible d'exécuter la requête SQL suivante

[checkEcrituresTrx.sql](#)

```
SELECT a.wallet_id, a.id, a.carte_emettrice_id, a.carte_receptrice_id,
a.code_evenement,a.code_utilisateur_saisie,a.date_enregistrement_transa
ction,a.date_transaction,
a.ind_annulation,a.is_online,a.montant_transaction,a.motif,a.motif_annu
lation,a.numero_operation,
a.operation_offline_id,a.produit,a.ref_terminal,a.report_cumul_consomma
tion_offline,
a.report_cumul_consommation_synchronise,a.report_plafond_offline,a.repo
rt_solde_online_on_card,
a.solde_apres_emmission,a.solde_apres_reception,a.solde_avant_emission,
```

```
a.solde_avant_reception,  
a.statut,a.uid_card,a.ticket_carburant_id,a.version_mobile,a.volume,  
a.statut_remontee_annulation,a.statut_synchro_to_etab,  
a.ind_espece,a.ind_trx_forcage_inter_oper,a.deleted+'0',  
b.num_carte AS numCarteEmetrice ,b.intitule AS  
intituleCarteEmetrice,b.statut AS 'statut_carte_emetrice',  
c.num_carte AS numCarteReceptrice ,c.intitule AS  
intituleCarteReceptrice,c.statut AS 'statut_carte_receptrice'  
FROM (operation a LEFT OUTER JOIN carte b ON a.carte_emettrice_id=b.id)  
LEFT OUTER JOIN carte c ON a.carte_receptrice_id=c.id  
WHERE a.id NOT IN (SELECT b.operation_id FROM ecriture b)  
AND a.date_transaction >= '2021-01-01'  
ORDER BY a.date_transaction ;
```

From:

<https://cynod.wiki.sensoft-next.com/> - cynod

Permanent link:

https://cynod.wiki.sensoft-next.com/doku.php?id=cynod:config_avancee:programmeregul

Last update: **2021/11/23 15:36**

