

Liste et description des requêtes SQL les plus utilisés sur CYNOD

Le but de cette documentation est de lister puis de commenter les différentes requêtes SQL qui sont les plus utilisés au sein sur CYNOD.

```
-- Récupérer le code Pin d'un client Wallet**
```

```
SELECT pin
FROM wallet
WHERE wallet_id=22791746657
```

```
-- Augmenter la durée de validité des cartes sur X années**
```

```
UPDATE carte c SET
user_update = "sensoft_aramé",
last_updated = Now(),
version = version + 1,
fin_validite = date_add(c.fin_validite , INTERVAL 2 YEAR)
WHERE
c.statut = "VA"
AND c.is_genere = FALSE
AND ind_corporate = FALSE;
```

```
-- Remettre le statut de bon de commande ticket a généré pour impression de la maquette PDF
```

```
<color #00a2e8>UPDATE bon_commande b
SET b.statut = "G",
b.version = b.version + 1,
b.user_update = "sensoft_mamadou",
b.last_updated =now()
WHERE
b.reference LIKE "%2104112023070223%";</color>
```

```
-- Listes des notifications non délivrées par SMSGW
```

```
SELECT
c.version, c.code_etatsms,c.cout_service_sms,
c.com_message_indiv_source_id,c.date_envoi_sms,
date_message,c.cout_service_sms, c.date_created,last_updated
,c.libelle_service,c.numero_destinataire,c.sender_etablissement_code,c.source,c.type_service, c.max_sms_resend_retry,country_code ,c.flag_envoi_sms
+"0",c.flag_traitement_sms +"0", c.ind_service_sms_facture
+"0",c.ind_service_sms_paye +"0", c.sms_resend_retry +"0"
FROM
com_message_indiv_web_service c
WHERE
```

```
date_created >="2022-11-18 18:00:00"  
AND flag_envoi_sms = 0;
```

```
-- Lever l'opposition ticket carburant expiré
```

```
UPDATE  
ticket_carburant  
SET  
last_updated = now(),  
comptabilise = 0,  
fin_validation = "2023-02-15 23:59:59",  
user_update = "m.wade",  
version = version + 1  
WHERE  
numero BETWEEN "2000640000000001" AND "2000640000001846"  
AND statut = "V";
```

```
-- Lever l'opposition des tickets carburant valides et archivés
```

Etape 1- Transférer les tickets archivés depuis la TABLE historique vers la TABLE principale ticket carburant

```
INSERT INTO ticket_carburant  
(id,  
version,  
comptabilise,  
date_created,  
date_validation,  
deleted,  
etablissement_id,  
fin_validation,  
last_updated,  
lot_ticket_detail_id,  
montant,  
numero,  
qr_code,  
service_point_consommation,  
statut,  
user_consommation,  
user_create,  
user_update,  
user_validation,  
volume,  
code_secret,  
opposition_ticket_id)  
(SELECT  
id,  
version,  
comptabilise,  
date_created,  
date_validation,
```

```
deleted,  
etablissement_id,  
fin_validation,  
last_updated,  
lot_ticket_detail_id,  
montant,  
numero,  
qr_code,  
service_point_consommation,  
statut,  
user_consommation,  
user_create,  
user_update,  
user_validation,  
volume,  
code_secret,  
opposition_ticket_id  
FROM historique_ticket_carburant  
WHERE numero BETWEEN "822001960000048586"  
AND "822001960000048785";
```

Etape 2- Suppression des tickets carburant sur la TABLE historique

```
DELETE FROM  
historique_ticket_carburant  
WHERE  
numero BETWEEN "822001960000048586"  
AND "822001960000048785";
```

Etape 3- Lever l'opposition des tickets carburant

```
update ticket_carburant set  
statut = "V",  
last_updated = now(),  
comptabilise = 0,  
user_update = "mindinga",  
fin_validation = "2023-02-15 23:59:59",  
version = version + 1  
where  
numero between "822001960000048586" and "822001960000048785"  
and statut = "X"  
and opposition_ticket_id is not NULL;
```

-- Mettre à jour la matrice de version

```
INSERT INTO matrice_version  
(version,  
action,  
date_created,  
date_version_mobile,  
date_version_serveur,
```

```
deleted,  
etablissement_id,  
last_updated,  
user_create,  
user_update,  
version_mobile,  
version_serveur)  
VALUES  
(1,  
"OK",  
now(),  
now(),  
now(),  
FALSE,  
1,  
now(),  
"amadou.diop",  
"amadou.diop",  
"3.1.16",  
"3.6.2_4");
```

-- Afficher la liste des notifications PIN générés côté B0 sur une période

```
SELECT  
c.date_depot, c.date_traitement, c.ind_msg_email+"0", c.ind_msg_releve+"0",  
c.ind_msg_valide+"0",  
c.ind_traitement+"0", cm.is_email_sent+"0", cm.date_envoi, cm.is_sms_sent+"0", c  
m.date_created, cm.date_envoi_sms, cm.flag_envoi_sms+"0",  
flag_envoi_email+"0", cm.ws_error_code, cm.ws_error_code_sms, cm.ws_error_code_  
mail,  
cm.code_etatsms, c.objet, u.email, u.tel_mobile  
FROM  
com_message c, com_message_indiv cm, USER u  
WHERE  
c.date_created BETWEEN '2022-10-24' AND '2022-10-29'  
AND cm.com_message_id = c.id  
AND cm.destinataire_individuel_id = u.id  
AND c.objet LIKE "%pin%";
```

-- Prolonger la date de fin de validité d'une carte

Etape 1- Lister des cartes expirant sur une période

```
SELECT c.intitule, c.num_carte, c.solde_online, c.debut_validite,  
c.fin_validite  
FROM carte c  
WHERE  
c.fin_validite >= STR_TO_DATE('01/01/2021 00:00', '%d/%m/%Y %H:%i') AND  
c.fin_validite <= STR_TO_DATE('31/12/2021 23:59', '%d/%m/%Y %H:%i')  
AND c.ind_annulation = FALSE  
AND c.statut = "VA"
```

```
AND c.is_genere = FALSE
AND c.ind_corporate = FALSE ORDER BY c.fin_validite,c.intitule DESC;
```

Etape 2- Mettre à jour les cartes expirant sur une période

```
UPDATE carte c SET user_update = "sensoft_mamadou",
last_updated = Now(),
version = version + 1,
fin_validite = '2022-12-31 23:59:59',
ind_annulation = FALSE
WHERE fin_validite >= STR_TO_DATE('01/01/2022 00:00', '%d/%m/%Y %H:%i') AND
fin_validite <= STR_TO_DATE('30/06/2022 23:59', '%d/%m/%Y %H:%i')
AND c.statut = "VA"
AND c.is_genere = FALSE
AND ind_corporate = FALSE;</color>
```

-- Lever opposition des cartes mises en opposition automatiquement et prolonger la date de fin de validité

Etape 1- Lister des cartes expirées sur la période

```
CREATE TABLE tmp_carte_levee_opposition AS
SELECT *
FROM carte c
WHERE fin_validite >= "2021-01-01 00:00:00" AND fin_validite <= "2021-12-31
23:59:59" AND is_point_service = FALSE
AND c.deleted = FALSE
AND c.is_genere = FALSE
AND c.ind_annulation = TRUE
AND c.ind_corporate = FALSE;
```

Etape 2- Mettre à jour les cartes expirées

```
UPDATE tmp_carte_levee_opposition tmp, carte c
SET c.user_update = "sensoft_mamadou",
c.last_updated = Now(),
c.version = c.version +1,
c.fin_validite = '2024-12-31 23:59:59',
c.statut="VA",
c.ind_annulation = FALSE
WHERE c.id = tmp.id;</color>
```

Etape 3- Transférer les cartes depuis la TABLE historique vers la TABLE carte

```
INSERT INTO opposition_carte
(`version`,
`carte_id`,
`date_created`,
`date_opposition`,
`deleted`,
```

```

`etablissement_id`,
`ind_opposition`,
`last_updated`,
`motif`,
`opposition`,
`user_create`,
`utilisateur_saisi`,
`from_portal`)

SELECT
op.`version`,
op.`carte_id`,
NOW(),
NOW(),
op.`deleted`,
op.`etablissement_id`,
0,
NOW(),
op.`motif`,
op.`opposition`,
"sensoft_mamadou",
"sensoft_mamadou",
op.`from_portal`
FROM
    opposition_carte op,
    tmp_carte_levee_opposition tmp
WHERE op.carte_id = tmp.id;

```

Etape 4- Mettre à jour la colonne `opposition_carte_id` de la TABLE `carte`

```

SET SQL_SAFE_UPDATES = 0;
UPDATE carte c
SET c.opposition_carte_id = (SELECT o.id FROM opposition_carte o WHERE
o.carte_id = c.id ORDER BY o.date_created DESC LIMIT 1)
WHERE c.deleted = FALSE;

```

Etape 5- Supprimer la TABLE temporaire

```

DROP TABLE tmp_carte_levee_opposition;

```

-- Rendre une carte mise en opposition automatiquement visible sur la fonctionnalité levée opposition

```

UPDATE carte
SET
version = version + 1,
last_updated = NOW(),
user_update = "sensoft_mamadou",
ind_annulation = FALSE,
fin_validite = "2025-01-13"
WHERE num_carte IN ("5812340100034468", "5812340100034088");

```

-- Lever opposition d'une carte mise en opposition automatiquement et prolonger la date de fin de validité

Etape 1- Mettre à jour les cartes expirant sur une période

```
UPDATE carte c
SET c.user_update = "GUELADIO BA",
c.last_updated = Now(),
c.version = c.version + 1,
c.fin_validite = '2029-12-31 23:59:59',
c.statut="VA",
c.ind_annulation = FALSE
WHERE
c.num_carte ="5812340100041448";
```

Etape 2- Transférer la carte depuis la TABLE historique vers la TABLE carte

```
INSERT INTO opposition_carte(
version,
carte_id,
date_created,
date_opposition,
deleted,
etablissement_id,
ind_opposition,
last_updated,
motif,
opposition,
user_create,
utilisateur_saisi,
from_portal)

SELECT
op.version,
op.carte_id,
NOW(),
NOW(),
op.deleted,
op.etablissement_id,
0,
NOW(),
op.motif,
op.opposition,
"GUELADIO BA",
"GUELADIO BA",
op.from_portal
FROM
    opposition_carte op,
    carte c
WHERE op.carte_id = c.id
```

```
AND c.num_carte ="5812340100041448";
```

Etape 3- Mettre à jour la colonne `opposition_carte_id` de la TABLE `carte`

```
UPDATE carte c
SET c.opposition_carte_id = (SELECT o.id FROM opposition_carte o WHERE
o.carte_id = c.id ORDER BY o.date_created DESC LIMIT 1)
WHERE
c.num_carte ="5812340100041448";
```

-- Lister les opérations n'ayant pas générées d'écriture

```
SELECT
o.numero_operation,
o.date_transaction,
o.date_created,
o.code_evenement,
o.montantttc,
o.solde_avant_emission,
o.solde_apres_emmission,
o.solde_avant_reception,
o.solde_apres_reception,
o.motif,
o.ind_annulation+"0",
o.is_online + "0",
c.num_carte,
c.intitule,
c.fin_validite AS "fin valite carte"
FROM operation o, carte c
WHERE o.numero_operation
AND o.statut = 'VA'
AND o.deleted = FALSE
AND o.carte_emettrice_id = c.id
AND c.statut = 'VA'
AND o.id NOT IN (SELECT ecriture.operation_id FROM ecriture)
ORDER BY o.date_created DESC, c.intitule;
```

-- Virtualiser une carte

```
* UPDATE carte SET statut = 'VA' WHERE num_carte = '5812340101000260';
* UPDATE lot_carte SET statut = 'LV' WHERE numero_lot = '3510';
```

-- Connaître le numéro de lot d'une carte

```
SELECT l.numero_lot
FROM lot_carte l, carte c
WHERE c.num_carte ="5812340100006805" AND c.lot_carte_id=l.id;
```

-- Permettre d'encoder à nouveau une carte

```
* UPDATE carte SET statut = 'IM' WHERE num_carte="5812340100049854";
```

```
* UPDATE lot_carte SET statut = 'IM' WHERE numero_lot = "1810";
```

```
-- Permettre d'imprimer à nouveau une carte
```

```
* UPDATE carte SET statut = 'SA' WHERE num_carte="5812340100049854";
* UPDATE lot_carte SET statut = 'VA' WHERE numero_lot = "1810";
```

```
-- Affecter une carte prépayée à un autre point de service
```

```
UPDATE
```

```
  miseadispotion_carte_carte mcc,
  miseadispotion_carte mc,
  carte c
  SET    mc.point_service_id=34
```

```
WHERE
```

```
  mcc.miseadispotion_carte_cartes_id = mc.id
  AND mcc.carte_id = c.id
  AND c.id IN (SELECT
    id
  FROM
    carte
  WHERE
    num_carte = "3538000000000000");
```

```
-- Supprimer tous les rapports jasper de CYNOD pour faire une nouvelle MAJ
```

```
* SET foreign_key_checks=0;
* TRUNCATE jasper_report_unit_resource;
* TRUNCATE jasper_report_unit_resource_role;
* SET foreign_key_checks=1;
```

```
-- Supprimer l'enrolement Wallet d'un client sur Cynod
```

Etape 1- Récupération de l'ID du compte Wallet avant suppression

```
select id from wallet w where wallet_id = "221775784470";
```

NB : Il est important de bien noté cet id qui va servir à l'étape 7 <code sql> Etape 2- Désactivation contrainte SET FOREIGN_KEY_CHECKS=0; </code> <code sql> Etape 3- Suppression du wallet_terminal DELETE wallet_terminal FROM wallet_terminal INNER JOIN wallet ON wallet.id = wallet_terminal.wallet_id WHERE wallet.id in (select id from wallet where wallet_id ="22792544040"); </code> <code sql> Etape 4- Suppression du wallet_carte DELETE wallet_carte FROM wallet_carte INNER JOIN wallet ON wallet.id = wallet_carte.wallet_id WHERE wallet.wallet_id in (select id from wallet where wallet_id ="22792544040"); </code> <code sql> Etape 5- Suppression du compte wallet delete from wallet where wallet_id="22792544040"; </code> <code sql> Etape 6- Activation contrainte SET FOREIGN_KEY_CHECKS=1; </code> <code sql> Etape 7- Mise à jour des opérations wallet suite à la nouvelle activation du client # 1 update historique_operation set wallet_id = (select id from wallet w where wallet_id = "221775784470") where wallet_id = 4 or wallet_receiver_id = 4; #2 update operation set wallet_id = (select id from wallet w where wallet_id = "221775784470") where wallet_id = 4 or wallet_receiver_id =

4; NB : Merci de remplacer les valeurs du wallet_id et du wallet_receiver_id par l'id récupéré à l'étape 1**

</code>

```
-- Rechercher l'existence d'une transaction sur Cynod
SELECT numero_operation_offline AS "numero", "Request Synchro Trx Offline"
FROM request_synchro_trx_offline WHERE numero_operation_offline IN
('6201-0006-220807230912', '6201-0006-220807230720')
UNION
SELECT numero_operation_offline AS "numero",
"Operation Offline Synchronised" AS tablename FROM
operation_offline_synchronized WHERE numero_operation_offline IN
('6201-0006-220807230912', '6201-0006-220807230720')
UNION
SELECT numero_operation_offline AS "numero", "Numero Operation Offline" AS
"tablename" FROM operation_offline_synchronized_history WHERE
numero_operation_offline IN ('6201-0006-220807230912',
'6201-0006-220807230720')
UNION
SELECT numero_operation_offline AS "numero", "Operation Offline Failed
Synchronised" AS "tablename" FROM operation_offline_failed_synchronized
WHERE numero_operation_offline IN ('6201-0006-220807230912',
'6201-0006-220807230720')
UNION
SELECT numero_operation AS "numero", "Anomalie Carte" AS "tablename" FROM
anomalies_carte WHERE numero_operation IN ('6201-0006-220807230912',
'6201-0006-220807230720')
UNION
SELECT numero_operation AS "numero", "Historique Oepration" AS "tablename"
FROM historique_operation WHERE numero_operation IN
('6201-0006-220807230912', '6201-0006-220807230720')
UNION
SELECT numero_operation AS "numero", "Operation" AS "tablename" FROM
operation WHERE numero_operation IN ('6201-0006-220807230912',
'6201-0006-220807230720');
```

```
-- Désactiver restriction Mot de passe expiré
UPDATE USER u SET
password_expired = FALSE,
u.user_update = "sensoft_aramé",
u.last_updated = now(),
u.version = u.version + 1
WHERE
u.username = "sensoft_ibrahima";
```

```
-- Mise à jour date erronée des transactions

### Transactions archivées
UPDATE historique_ecriture e, historique_operation o
SET e.date_ecriture = "2024-02-08 22:40:00",
e.last_updated = NOW(),
```

```
e.user_update = "djibril.traore",
e.version = e.version +1
WHERE e.numero_operation=o.numero_operation
AND o.numero_operation IN
('5806-0043-700102202211', '5806-0043-700102202530');

UPDATE historique_operation o, historique_ecriture e
SET o.date_transaction = e.date_ecriture,
o.last_updated = NOW(),
o.user_update = "djibril.traore",
o.version = o.version +1
WHERE o.numero_operation = e.numero_operation
AND o.numero_operation IN
('5806-0043-700102202211', '5806-0043-700102202530');

### Transactions non archivées
UPDATE ecriture e, operation o
SET e.date_ecriture = "2024-02-08 22:40:00",
e.last_updated = NOW(),
e.user_update = "djibril.traore",
e.version = e.version +1
WHERE e.operation_id =o.id
AND o.numero_operation IN
('5806-0043-700102202211', '5806-0043-700102202530');

UPDATE operation o, ecriture e
SET o.date_transaction = e.date_ecriture,
o.last_updated = NOW(),
o.user_update = "djibril.traore",
o.version = o.version +1
WHERE o.id = e.operation_id
AND o.numero_operation IN
('5806-0043-700102202211', '5806-0043-700102202530');
```

From:

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

Permanent link:

<https://cynod.wiki.sensoft-next.com/doku.php?id=tecdoc:requetesql&rev=1739189128>

Last update: **2025/02/10 12:05**

