

Database Queries to handle coins and subscriptions

In the coin_vault table, directly modifying the coin_balance using manual queries will update the balance for a given user. However, this change alone does not allow the user to create Treels, as the coin_vault, subscription, and payment tables are interlinked. The is_subscription_active flag is derived by validating whether the user's subscription is still active or expired. This is determined by comparing the payment_date with the user's chosen subscription plan and calculating the remaining validity period. Therefore, to properly activate a user and enable Treel creation, it is necessary to update not only the coin_balance but also the corresponding payment_date and subscription details. This ensures that the subscription status reflects an active state and aligns with the system's business logic.

Query to get user id:

```
select * from app_user where email_id='example@email.com'
```

Query to check the coin balance of a user:

```
select * from coin_vault where app_user_id=123
```

Query to add the coins to the coin_vault:

```
UPDATE coin_vault
```

```
SET coin_balance = 500
```

```
WHERE app_user_id = 123;
```

Query to check whether the user subscription active or not:

```
WITH coin_balance AS ( -- coin balance has to be > 0
```

```
    SELECT app_user_id, coin_balance
```

```
    FROM public.coin_vault
```

```
    WHERE app_user_id = 123
```

```
),
```

```
latest_payment AS ( -- Check latest payment to make sure the subscription is active
```

```
    SELECT DISTINCT ON (sp.user_subscription_id)
```

```
        sp.user_subscription_id,
```

```

        sp.payment_date
FROM public.subscription_payment sp
WHERE sp.status = 'PAID'
ORDER BY sp.user_subscription_id, sp.payment_date DESC
),
series_counts AS ( -- Check if any series limits for the subscribed plan
SELECT
    nsr.app_user_id,
    sp.user_subscription_id,

    -- Series created today
    COUNT(*) FILTER (WHERE DATE(nsr.created_dttm) = CURRENT_DATE) AS
series_created_today,

    -- Series created in the last recurring billing day (monthly/yearly)
    COUNT(*) FILTER (
WHERE nsr.created_dttm >=
CASE
    WHEN us.billing_cycle = 'MONTHLY' THEN sp.payment_date
    WHEN us.billing_cycle = 'YEARLY' THEN
        sp.payment_date
        + INTERVAL '1 month' * FLOOR(
            EXTRACT(YEAR FROM AGE(CURRENT_DATE, sp.payment_date)) * 12
            + EXTRACT(MONTH FROM AGE(CURRENT_DATE, sp.payment_date))
        )

```

```

        ELSE CURRENT_DATE - INTERVAL '1 month' -- fallback
    END

    AND nsr.created_dttm < CURRENT_DATE

) AS series_created_this_month,

-- Series created in the 1-year window since payment
COUNT(*) FILTER (

    WHERE nsr.created_dttm >= sp.payment_date AND nsr.created_dttm <
sp.payment_date + INTERVAL '1 year'

) AS series_created_this_year

FROM public.new_series_request nsr

JOIN public.user_subscription us ON nsr.app_user_id = us.app_user_id

JOIN latest_payment sp ON sp.user_subscription_id = us.user_subscription_id

WHERE nsr.requested_dttm >= sp.payment_date

AND nsr.app_user_id = 123

GROUP BY nsr.app_user_id, sp.user_subscription_id, us.billing_cycle,
sp.payment_date

)

SELECT

    us.user_subscription_id,

    us.app_user_id,

    COALESCE(cb.coin_balance, 0) AS coin_balance,

-- Subscription Type Logic

CASE

    WHEN us.coin_count > 0

```

```

        AND COALESCE(us.daily_new_series_limit, 0) = 0
        AND COALESCE(us.monthly_new_series_limit, 0) = 0
        AND COALESCE(us.yearly_new_series_limit, 0) = 0
        AND us.billing_cycle = 'MONTHLY' THEN 'MONTHLY_COIN_LIMIT'
    WHEN us.coin_count > 0
        AND COALESCE(us.daily_new_series_limit, 0) = 0
        AND COALESCE(us.monthly_new_series_limit, 0) = 0
        AND COALESCE(us.yearly_new_series_limit, 0) = 0
        AND us.billing_cycle = 'YEARLY' THEN 'YEARLY_COIN_LIMIT'
    WHEN COALESCE(us.daily_new_series_limit, 0) > 0
        OR COALESCE(us.monthly_new_series_limit, 0) > 0
        OR COALESCE(us.yearly_new_series_limit, 0) > 0 THEN 'TREEL_LIMIT'
    ELSE 'FREE'
END AS subscription_type,

lp.payment_date,

-- Active subscription logic
CASE
    WHEN us.billing_cycle = 'MONTHLY' AND lp.payment_date >= CURRENT_DATE -
INTERVAL '1 month' THEN TRUE
    WHEN us.billing_cycle = 'YEARLY' AND lp.payment_date >= CURRENT_DATE -
INTERVAL '1 year' THEN TRUE
    ELSE FALSE
END AS is_subscription_active,

us.daily_new_series_limit,

```

```

us.monthly_new_series_limit,
us.yearly_new_series_limit,

-- Series stats
COALESCE(sc.series_created_today, 0) AS series_created_today,
COALESCE(sc.series_created_this_month, 0) AS series_created_this_month,
COALESCE(sc.series_created_this_year, 0) AS series_created_this_year
FROM public.user_subscription us
LEFT JOIN latest_payment lp ON lp.user_subscription_id = us.user_subscription_id
LEFT JOIN series_counts sc ON sc.user_subscription_id = us.user_subscription_id
LEFT JOIN coin_balance cb ON cb.app_user_id = us.app_user_id
WHERE us.is_active = TRUE
AND us.payment_status = 'PAID'
AND COALESCE(us.coin_count, 0) > 0
AND CURRENT_DATE BETWEEN us.start_date AND COALESCE(us.end_date,
CURRENT_DATE)
AND us.app_user_id = 123;

```

Query to make the subscription active and change payment date to current date:

```

UPDATE public.subscription_payment
SET status = 'PAID',
    payment_date = CURRENT_DATE
WHERE user_subscription_id = (
    SELECT user_subscription_id
    FROM public.user_subscription
    WHERE app_user_id = 123
);

```

Curls used to activate the channel and subscription of user through backend

- 1) Run this curl first (change the app user id and date to the current date)

```
curl --location 'http://localhost:8080/api/v1/razorpay/user-subscription/add' \
--header 'Content-Type: application/json' \
--data '{
  "appUserId": 123,
  "billingCycle": "YEARLY",
  "subscriptionPlanId": 10,
  "coinCount": 20000,
  "subscriptionPrice": 2999,
  "paymentStatus": "PENDING",
  "dailyNewSeriesLimit": 0,
  "monthlyNewSeriesLimit": 0,
  "yearlyNewSeriesLimit": 0,
  "isActive": true,
  "startDate": "2025-09-10",
  "razorpayCustomerId": "cust_Qwsv3vXncGD38N",
  "razorpaySubscriptionId": "sub_Qwsv66wv4KkFbu"
}'
```

- 2) Copy the usersubscriptionId from the response by running the first curl and paste it in the second curl given below

```
curl --location 'http://localhost:8080/api/v1/razorpay/subscription-payment/add' \
--header 'Content-Type: application/json' \
--data '{
  "amount": 2999,
  "createdUser": "system",
  "currency": "INR",
```

```
"paidAt": "2025-09-09T10:10:41.1592887",  
"paymentGateway": "RAZORPAY",  
"razorpayPaymentId": "string",  
"receipt": "sub_Qwm34HhrOtQubG",  
"status": "PAID",  
"userSubscriptionId": 123  
}  
,
```