```
from unittest.mock import Mock


import pytest

from supabase import Client

from swarms_cloud.loggers.supabase_handler import SwarmCloudUsageLogger



# Create a mock Supabase client for testing

@pytest.fixture

def mock_supabase_client():

    return Mock(spec=Client)



# Test cases for SwarmCloudUsageLogger class

def test_swarm_cloud_usage_logger_initialization():

    logger = SwarmCloudUsageLogger(

        supabase_url="https://your_supabase_url.supabase.co",

        supabase_key="your_supabase_key",

    )

    assert isinstance(logger, SwarmCloudUsageLogger)



def test_swarm_cloud_usage_logger_log_usage(mock_supabase_client):

    logger = SwarmCloudUsageLogger(

        supabase_url="https://your_supabase_url.supabase.co",

        supabase_key="your_supabase_key",
    )
    logger.supabase = mock_supabase_client
    response = logger.log_usage(
        user_id="user123",
        api_key="sk-your_api_key",
        usage_type="get",
        tokens_used=1,
    )
    assert response is not None


def test_swarm_cloud_usage_logger_log_usage_error(mock_supabase_client):
    logger = SwarmCloudUsageLogger(
        supabase_url="https://your_supabase_url.supabase.co",
        supabase_key="your_supabase_key",
    )
    logger.supabase = mock_supabase_client
    logger.supabase.table(
        "swarm_cloud_usage"
    ).insert.return_value.execute.side_effect = Exception("Test Exception")
    with pytest.raises(RuntimeError):
        logger.log_usage(
            user_id="user123",
            api_key="sk-your_api_key",
            usage_type="get",
            tokens_used=1,
        )



def test_swarm_cloud_usage_logger_check_api_key(mock_supabase_client):
    logger = SwarmCloudUsageLogger(
        supabase_url="https://your_supabase_url.supabase.co",
        supabase_key="your_supabase_key",
    )
    logger.supabase = mock_supabase_client
    mock_supabase_client.table(
        "swarm_cloud_usage"
    ).select.return_value.eq.return_value.execute.return_value.data = [
        {"api_key": "sk-your_api_key"}
    ]
    result = logger.check_api_key(api_key="sk-your_api_key")
    assert result is True



def test_swarm_cloud_usage_logger_check_api_key_not_found(mock_supabase_client):
    logger = SwarmCloudUsageLogger(
        supabase_url="https://your_supabase_url.supabase.co",
        supabase_key="your_supabase_key",
    )
    logger.supabase = mock_supabase_client
    mock_supabase_client.table(
        "swarm_cloud_usage"
    ).select.return_value.eq.return_value.execute.return_value.data = []
    result = logger.check_api_key(api_key="sk-your_api_key")
    assert result is False



def test_swarm_cloud_usage_logger_check_api_key_error(mock_supabase_client):
    logger = SwarmCloudUsageLogger(
        supabase_url="https://your_supabase_url.supabase.co",
        supabase_key="your_supabase_key",
    )
    logger.supabase = mock_supabase_client
    mock_supabase_client.table(
        "swarm_cloud_usage"
    ).select.return_value.eq.return_value.execute.side_effect = Exception(
        "Test Exception"
    )
    with pytest.raises(RuntimeError):
        logger.check_api_key(api_key="sk-your_api_key")
```