### Criar tabelas

Abaixo seguem os comandos create da tabela de cartão de crédito e de sua tabela de log.

Abaixo de cada comando podemos conferir a evidência de execução.

CREATE TABLE t\_mc\_cartao\_credito\_cli

(

id\_cartao\_credito *NUMBER*(10) NOT NULL,

nr\_cartao\_credito *VARCHAR2*(20) NOT NULL,

nm\_cliente\_cartao *VARCHAR2*(100) NOT NULL,

nm\_bandeira\_cartao *VARCHAR2*(50) NOT NULL,

mes\_ano\_vencto *VARCHAR2*(6) NOT NULL,

nr\_codigo\_seguranca *VARCHAR2*(3) NOT NULL,

CONSTRAINT pk\_t\_mc\_cartao\_credito\_cli PRIMARY KEY (id\_cartao\_credito)

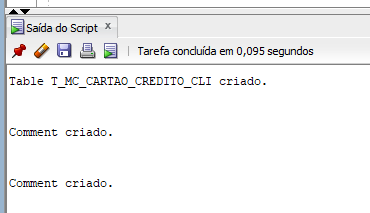
);

COMMENT ON TABLE t\_mc\_cartao\_credito\_cli IS

'Tabela de cartões de crédito dos clientes';

COMMENT ON COLUMN t\_mc\_cartao\_credito\_cli.id\_cartao\_credito IS

'Identificador único do cartão de crédito';



CREATE TABLE t\_mc\_cartao\_credito\_log

(

id\_log *NUMBER*(10) NOT NULL,

dt\_movimentacao *TIMESTAMP* NOT NULL,

nm\_operacao\_uid *VARCHAR2*(6) NOT NULL,

id\_cartao\_credito *NUMBER*(10),

nr\_cartao\_credito\_novo *VARCHAR2*(20),

nr\_cartao\_credito\_antigo *VARCHAR2*(20),

nm\_cliente\_cartao\_novo *VARCHAR2*(100),

nm\_cliente\_cartao\_antigo *VARCHAR2*(100),

nm\_bandeira\_cartao\_novo *VARCHAR2*(50),

nm\_bandeira\_cartao\_antigo *VARCHAR2*(50),

mes\_ano\_vencto\_novo *VARCHAR2*(6),

mes\_ano\_vencto\_antigo *VARCHAR2*(6),

nr\_codigo\_seguranca\_novo *VARCHAR2*(3),

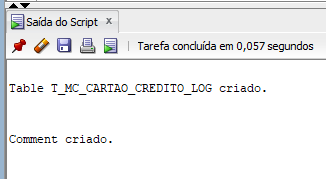
nr\_codigo\_seguranca\_antigo *VARCHAR2*(3),

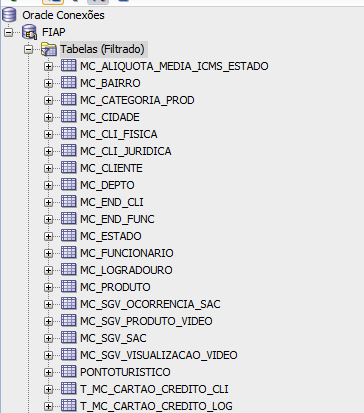
CONSTRAINT pk\_t\_mc\_cartao\_credito\_log PRIMARY KEY (id\_log)

);

COMMENT ON TABLE t\_mc\_cartao\_credito\_log IS

'Tabela de auditoria de cartões de crédito (controlada por trigger)';





### Criar Triggers

A trigger **dispara antes** (BEFORE) de cada operação INSERT, UPDATE ou DELETE (ou seja, FOR EACH ROW) na tabela de origem, registrando o histórico na tabela de log:

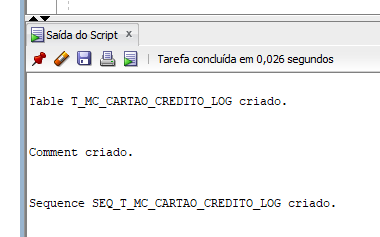
CREATE SEQUENCE seq\_t\_mc\_cartao\_credito\_log

START WITH 1

INCREMENT BY 1

NOCACHE

NOCYCLE;



CREATE OR replace TRIGGER tr\_t\_mc\_cartao\_credito\_cli\_log

BEFORE INSERT OR UPDATE OR DELETE ON t\_mc\_cartao\_credito\_cli

FOR EACH ROW

DECLARE

v\_operacao *VARCHAR2*(6);

BEGIN

*-- Descobre o tipo de operação*

IF inserting THEN

v\_operacao := 'INSERT';

ELSIF updating THEN

v\_operacao := 'UPDATE';

ELSIF deleting THEN

v\_operacao := 'DELETE';

END IF;

INSERT INTO t\_mc\_cartao\_credito\_log

(id\_log,

dt\_movimentacao,

nm\_operacao\_uid,

id\_cartao\_credito,

nr\_cartao\_credito\_novo,

nr\_cartao\_credito\_antigo,

nm\_cliente\_cartao\_novo,

nm\_cliente\_cartao\_antigo,

nm\_bandeira\_cartao\_novo,

nm\_bandeira\_cartao\_antigo,

mes\_ano\_vencto\_novo,

mes\_ano\_vencto\_antigo,

nr\_codigo\_seguranca\_novo,

nr\_codigo\_seguranca\_antigo)

VALUES (

*-- 1) ID\_LOG gerado pela sequence*

seq\_t\_mc\_cartao\_credito\_log.NEXTVAL,

*-- 2) Data/hora do evento*

systimestamp,

*-- 3) Nome da operação (INSERT, UPDATE ou DELETE)*

v\_operacao,

*-- 4) ID\_CARTAO\_CREDITO*

CASE v\_operacao

WHEN 'DELETE' THEN :OLD.id\_cartao\_credito

ELSE :NEW.id\_cartao\_credito

END,

*-- 5) NR\_CARTAO\_CREDITO\_NOVO*

CASE v\_operacao

WHEN 'INSERT' THEN :NEW.nr\_cartao\_credito

WHEN 'UPDATE' THEN :NEW.nr\_cartao\_credito

ELSE NULL

END,

*-- 6) NR\_CARTAO\_CREDITO\_ANTIGO*

CASE v\_operacao

WHEN 'INSERT' THEN NULL

WHEN 'UPDATE' THEN :OLD.nr\_cartao\_credito

WHEN 'DELETE' THEN :OLD.nr\_cartao\_credito

END,

*-- 7) NM\_CLIENTE\_CARTAO\_NOVO*

CASE v\_operacao

WHEN 'INSERT' THEN :NEW.nm\_cliente\_cartao

WHEN 'UPDATE' THEN :NEW.nm\_cliente\_cartao

ELSE NULL

END,

*-- 8) NM\_CLIENTE\_CARTAO\_ANTIGO*

CASE v\_operacao

WHEN 'INSERT' THEN NULL

WHEN 'UPDATE' THEN :OLD.nm\_cliente\_cartao

WHEN 'DELETE' THEN :OLD.nm\_cliente\_cartao

END,

*-- 9) NM\_BANDEIRA\_CARTAO\_NOVO*

CASE v\_operacao

WHEN 'INSERT' THEN :NEW.nm\_bandeira\_cartao

WHEN 'UPDATE' THEN :NEW.nm\_bandeira\_cartao

ELSE NULL

END,

*-- 10) NM\_BANDEIRA\_CARTAO\_ANTIGO*

CASE v\_operacao

WHEN 'INSERT' THEN NULL

WHEN 'UPDATE' THEN :OLD.nm\_bandeira\_cartao

WHEN 'DELETE' THEN :OLD.nm\_bandeira\_cartao

END,

*-- 11) MES\_ANO\_VENCTO\_NOVO*

CASE v\_operacao

WHEN 'INSERT' THEN :NEW.mes\_ano\_vencto

WHEN 'UPDATE' THEN :NEW.mes\_ano\_vencto

ELSE NULL

END,

*-- 12) MES\_ANO\_VENCTO\_ANTIGO*

CASE v\_operacao

WHEN 'INSERT' THEN NULL

WHEN 'UPDATE' THEN :OLD.mes\_ano\_vencto

WHEN 'DELETE' THEN :OLD.mes\_ano\_vencto

END,

*-- 13) NR\_CODIGO\_SEGURANCA\_NOVO*

CASE v\_operacao

WHEN 'INSERT' THEN :NEW.nr\_codigo\_seguranca

WHEN 'UPDATE' THEN :NEW.nr\_codigo\_seguranca

ELSE NULL

END,

*-- 14) NR\_CODIGO\_SEGURANCA\_ANTIGO*

CASE v\_operacao

WHEN 'INSERT' THEN NULL

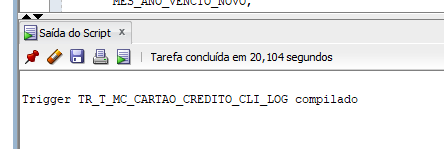
WHEN 'UPDATE' THEN :OLD.nr\_codigo\_seguranca

WHEN 'DELETE' THEN :OLD.nr\_codigo\_seguranca

END );

END;

/



### Comandos INSERT, UPDATE e DELETE

Faremos **3 inserts** na tabela de origem T\_MC\_CARTAO\_CREDITO\_CLI com dados fictícios:

INSERT INTO t\_mc\_cartao\_credito\_cli

(id\_cartao\_credito,

nr\_cartao\_credito,

nm\_cliente\_cartao,

nm\_bandeira\_cartao,

mes\_ano\_vencto,

nr\_codigo\_seguranca)

VALUES (1,

'1234 5678 9012 3456',

'JOÃO DA SILVA',

'VISA',

'072025',

'123');

INSERT INTO t\_mc\_cartao\_credito\_cli

(id\_cartao\_credito,

nr\_cartao\_credito,

nm\_cliente\_cartao,

nm\_bandeira\_cartao,

mes\_ano\_vencto,

nr\_codigo\_seguranca)

VALUES (2,

'1111 2222 3333 4444',

'MARIA DE SOUZA',

'MASTERCARD',

'082025',

'456');

INSERT INTO t\_mc\_cartao\_credito\_cli

(id\_cartao\_credito,

nr\_cartao\_credito,

nm\_cliente\_cartao,

nm\_bandeira\_cartao,

mes\_ano\_vencto,

nr\_codigo\_seguranca)

VALUES (3,

'9999 8888 7777 6666',

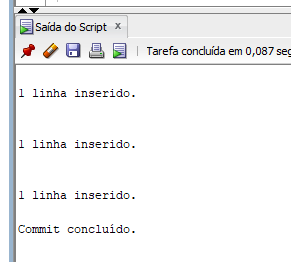
'CARLOS OLIVEIRA',

'ELO',

'102025',

'789');

COMMIT;



Cada INSERT dispara a trigger e grava o histórico na T\_MC\_CARTAO\_CREDITO\_LOG.

Faremos **3 updates**, alterando alguns dados:

*-- 1. Atualiza o nome e a bandeira do cartão de ID 1*

UPDATE t\_mc\_cartao\_credito\_cli

SET nm\_cliente\_cartao = 'JOÃO SILVA',

nm\_bandeira\_cartao = 'VISA GOLD'

WHERE id\_cartao\_credito = 1;

*-- 2. Atualiza o número do cartão do ID 2*

UPDATE t\_mc\_cartao\_credito\_cli

SET nr\_cartao\_credito = '0000 2222 3333 4444'

WHERE id\_cartao\_credito = 2;

*-- 3. Atualiza a data de vencimento e o código de segurança do ID 3*

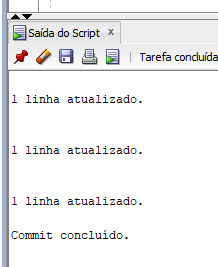
UPDATE t\_mc\_cartao\_credito\_cli

SET mes\_ano\_vencto = '122025',

nr\_codigo\_seguranca = '999'

WHERE id\_cartao\_credito = 3;

COMMIT;



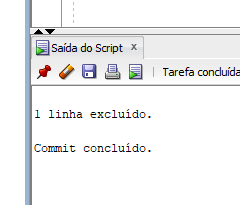
Cada UPDATE também dispara a trigger, gerando novos registros de histórico.

Faremos **1 delete** removendo o registro do ID 2, por exemplo:

DELETE FROM t\_mc\_cartao\_credito\_cli

WHERE id\_cartao\_credito = 2;

COMMIT;



O DELETE igualmente disparará a trigger, gravando o registro de auditoria.

—------------------------------------—-------------------------------------------------------------------------------------

### Checar Logs

Por fim, podemos consultar a tabela de log para verificar o resultado da auditoria:

SELECT id\_log,

dt\_movimentacao,

nm\_operacao\_uid,

id\_cartao\_credito,

nr\_cartao\_credito\_novo,

nr\_cartao\_credito\_antigo,

nm\_cliente\_cartao\_novo,

nm\_cliente\_cartao\_antigo,

nm\_bandeira\_cartao\_novo,

nm\_bandeira\_cartao\_antigo,

mes\_ano\_vencto\_novo,

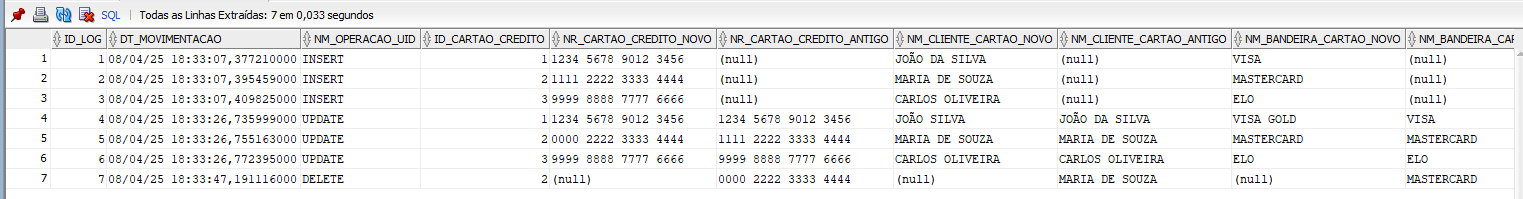
mes\_ano\_vencto\_antigo,

nr\_codigo\_seguranca\_novo,

nr\_codigo\_seguranca\_antigo

FROM t\_mc\_cartao\_credito\_log

ORDER BY id\_log;



Podemos observar:

* Linhas com NM\_OPERACAO\_UID = 'INSERT' para os três inserts.
* Linhas com NM\_OPERACAO\_UID = 'UPDATE' para os três updates.
* Linha com NM\_OPERACAO\_UID = 'DELETE' para o registro removido.

Além disso, cada linha mostrará os valores antigos e novos conforme as regras definidas na trigger.

### Dropar tabelas e triggers:

Abaixo os comandos de drop caso desejemos repetir o fluxo.

*-- Tabela de origem*

DROP TABLE t\_mc\_cartao\_credito\_cli CASCADE CONSTRAINTS purge;

*-- Tabela de log*

DROP TABLE t\_mc\_cartao\_credito\_log CASCADE CONSTRAINTS purge;

DROP SEQUENCE seq\_t\_mc\_cartao\_credito\_log;

DROP TRIGGER tr\_t\_mc\_cartao\_credito\_cli\_log;