

PROIECT SECURITATEA BAZELOR DE DATE

Realizat de: Olteanu-Crăciunescu Ana-Maria

Grupa 505, BDTS

Cuprins

1. Introducere	3
a. Prezentarea succintă a modelului prezentat si a regulilor sale	3
b. Diagrama conceptuală asociată modelului	4
c. Schemele relaționale asociate diagramei	4
d. Crearea tabelelor	5
e. Prezentarea regulilor de securitate care vor fi aplicate asupra modelului	5
2. Criptarea datelor	5
3. Auditarea activităților asupra bazei de date	10
a. Auditare standard	10
b. Trigger-i de auditare	13
c. Politici de auditare	17
4. Gestiunea utilizatorilor unei baze de date și a resurselor computaționale.....	20
a. Proiectarea configurației de management a identităților în baza de date (matricile proces-utilizator, entitate-proces, entitate-utilizator)	20
b. Implementarea configurației de management a identităților în baza de date	22
5. Privilegii si roluri.....	32
a. Privilegii sistem si obiect	32
6. Aplicațiile pe baza de date și securitatea datelor	37
a. Contextul aplicației	37
b. SQL Injection	40

1. Introducere

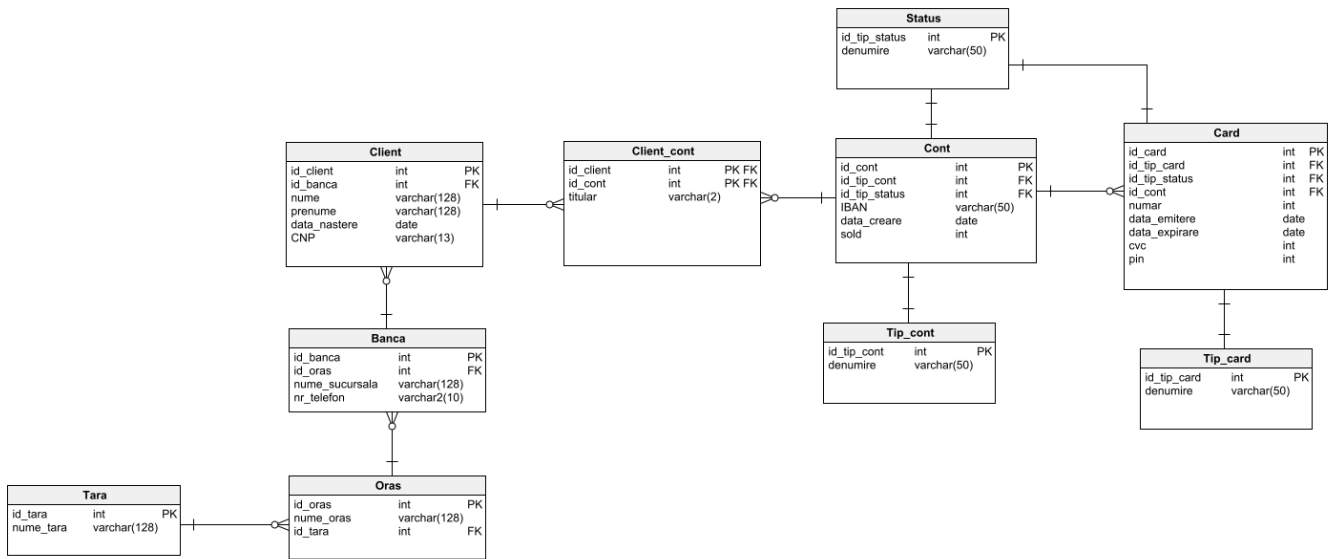
a. Prezentarea succintă a modelului prezentat si a regulilor sale

Modelul proiectat își propune să fie integrat într-un sistem intern de gestiune a unei bănci. Scopul acestui model este de a stoca și de a gestiona informații despre sucursalele unei bănci, despre clienții care și-au deschis conturi la una dintre sucursalele existente, precum și a cardurilor emise asociate conturilor. Acest model permite băncii să gestioneze eficient și în siguranță informațiile clienților și să furnizeze servicii financiare de calitate. De acest sistem de gestiune se pot folosi atât angajații băncii, cât și clienții care doresc să afle informații despre conturile sau cardurile pe care le dețin.

Regulile ce au fost luate în considerare la momentul proiectării modelului vizează în principal entitățile banca (sucursalele unei bănci specifice), client, conturi și carduri. În continuare voi prezenta regulile după care a fost implementat modelul de baze de date pentru o bancă:

1. Pentru o sucursală a băncii se rețin informații precum numele, numărul de telefon și orașul în care se găsește
2. Pentru client se rețin informații precum nume, prenume, data nașterii și CNP-ul
3. Pentru un cont se vor stoca date sensibile precum IBAN-ul, soldul și data creării. Banca pune la dispoziție următoarele tipuri de conturi: curent, de economii și de salariu. Un cont poate aparține mai multor clienți, fiind specificat care dintre aceștia este titularul contului. Conturile se pot găsi fie în status „ACTIV”, fie în status „INACTIV”, în funcție de cerințele clientului.
4. Entitatea asociată cardului va reține informații sensibile, precum numărul cardului, data emiterii, data expirării, codul CVC și codul PIN stabilit de client. Pentru fiecare cont se pot asocia unul sau mai multe carduri, având tipurile: credit, debit, preplătit sau de masă.

b. Diagrama conceptuală asociată modelului



c. Schemele relaționale asociate diagramei

TARA(#id_tara, nume_tara)

ORAS(#id_oras, nume_oras, id_tara)

BANCA(#id_banca, nume_sucursala, nr_telefon, id_oras)

CLIENT(#id_client, nume, prenume, data_nastere, CNP, id_banca)

CLIENT_CONT(#id_client, #id_cont, titular)

CONT(#id_cont, IBAN, data_creare, sold, id_tip_cont, id_tip_status)

TIP_CONT(#id_tip_cont, denumire)

CARD(#id_card, numar, data_emitere, data_expirare, cvc, pin, id_tip_card, id_tip_status)

TIP_CARD(#id_tip_card, denumire)

STATUS(#id_tip_status, denumire)

d. Crearea tabelelor

Scripturile de creare a tabelelor se află într-un fișier separat

e. Prezentarea regulilor de securitate care vor fi aplicate asupra modelului

Asupra modelului implementat vor fi aplicate reguli de securitate precum:

- Criptarea datelor sensibile
- Auditarea activităților asupra bazei de date prin diverse metode
- Gestiunea utilizatorilor unei baze de date și a resurselor computaționale
- Acordarea de privilegii și roluri asupra sistemului și obiectelor
- Crearea unui context al aplicației
- Prevenirea atacurilor de tip SQL Injection

2. Criptarea datelor

Urmează a fi criptate datele sensibile pentru modelul proiectat. Pentru criptare se va folosi funcția ENCRYPT din pachetul *dbms_crypto*, care va utiliza ca algoritm de criptare AES128, cu padding PKCS5 și metoda de chaining ECB. Coloanele vizate pentru criptare sunt:

- IBAN din tabela CONT
- numar, cvc, pin din tabela CARD

Pentru fiecare tabel se va genera câte o cheie secretă, cu ajutorul funcției *randombytes()*. Cheile vor fi stocate într-un tabel separat, în aceeași bază de date, care are drept coloane: *id_cheie*, *cheie* și *nume_tabel*. Am creat o procedură prin care se inserează câte o cheie de criptare pentru fiecare dintre tabelele CONT, respectiv CARD.

```
create or replace procedure inserare_chei_criptare(p_nume_tabel varchar2)
as
    cheie_secreta raw(16);
    nr_chei NUMBER(5);
begin
    cheie_secreta := dbms_crypto.randombytes(16);

    select count(*) into nr_chei
```

```

from tabel_chei_banca
where UPPER(ume_tabel) = UPPER(p_ume_tabel);

if nr_chei = 0 then
    insert into tabel_chei_banca values(secv_id_cheie.nextval, cheie_secreta, p_ume_tabel);
end if;

end;
/

begin
    insertare_chei_criptare('CONT');
    insertare_chei_criptare('CARD');
end;
/

```

Procedure INSERTARE_CHEI_CRIPTARE compiled

PL/SQL procedure successfully completed.

All Rows Fetched: 2 in 0.001 seconds			
	ID_CHEIE	CHEIE	NUME_TABEL
1	1	EC663C7573B63557143A4131E63341AA	CONT
2	2	D997C091A03F5E5DA23FC12DDABC9545	CARD

Am creat două proceduri prin care voi cripta coloanele menționate anterior, câte o procedură pentru fiecare tabel în cauză.

```

create or replace procedure criptare_cont
as

```

```

cheie_secreta raw(16);
mod_operare pls_integer;
cursor c_cont is select iban, id_cont from cont;
iban_raw raw(100);
rezultat_iban raw(100);

begin

select cheie into cheie_secreta
from tabel_chei_banca
where UPPER(ume_tabel) = UPPER('CONT');

dbms_output.put_line('Cheia de criptare pt tabelul CONT este ' || cheie_secreta);

mod_operare := dbms_crypto.encrypt_aes128 + dbms_crypto.pad_pkcs5 +
dbms_crypto.chain_cbc;

for rec in c_cont loop
    iban_raw := utl_i18n.string_to_raw(rec.iban, 'AL32UTF8');
    if length(iban_raw) < 64 then
        rezultat_iban := dbms_crypto.encrypt(iban_raw, mod_operare, cheie_secreta);
        update cont
        set iban = rezultat_iban
        where id_cont = rec.id_cont;
    end if;
end loop;
commit;
end;
/

begin

```

```

    criptare_cont();
end;

/

```

Procedure CRIPTARE_CONT compiled

Cheia de criptare pt tabelul CONT este EC663C7573B63557143A4131E63341AA

PL/SQL procedure successfully completed.

ID_CONT	ID_TIP_CONT	ID_TIP_STATUS	IBAN	DATA_CREARE	SOLD
1	1	1	1 90A5C913AB71C3FC5E1DD7E2C1D4A74C11F1CA1B243EA2F6483D5DA24EE748EF	12-SEP-18	299
2	2	1	1 0B8CC2C62C9D020C0C041430DD0AC8EFDA4C606A46696C933009E3365A26C592	17-JAN-00	4681
3	3	1	2 9EC08B693902E24D6F560BDEDF5E4F3D62F3126E415BA25F8B5D2BDCA0EB81D4	18-NOV-02	6689
4	4	1	2 9933975456C6A805AE7947772B4147277A76886B8B250BD1F234A565A8566C6C	14-JUN-96	7020
5	5	2	1 AAE55E9A4C82851290A0AE02B4D3CC8CAE02FE0D69E618ED8A1B9D50F6A49166	17-AUG-04	2258
6	6	3	1 CC93AD03BAAA7449089DE1A3097D0DBCAF5A04E47B44604FF38E48FB1ED1C350	10-JAN-21	1580
7	7	1	1 64B78C2D280BD650C3FA3F54AD6DEF5890C4C619FA5FD0DFD9BA29C84884E5E6	13-JUL-97	8496
8	8	2	2 A3CCFAA4B83EDF99B535D14D25115F9492FFFD1B9921F45025CF1B7106239074	11-JUL-97	6877
9	9	3	2 B17229D39A4DA8B117F38EDFC1843559D6FCB151A017D3B05AF9B07CAEB23D18	26-AUG-04	1550
10	10	1	1 AC8B322355C4A97FF6734E70B7ED6567CF49B4A797B84BA47C81C70F365E3860	03-OCT-14	5505
11	11	2	1 24BE7C9AEE810F111FD3BB927DEB6F0543A579963AD11A46AB38C2B00A279294	13-JAN-04	3371
12	12	1	2 80D2D759265ED57C981BC720606B9EA69D17B93777A0DECA4659946DCDEC85F9	27-JAN-06	5653
13	13	3	2 8BC64F702E06B37828F539405261934E69DB5B3BED66BD6F65EF70F0E201A2CA	31-JAN-95	3569
14	14	1	2 D9EC6E235A74B3B07CC834DAAE1458DE4D0F90BDC935ABAB94A178E9689A52ED	20-JAN-07	7118
15	15	2	2 C442F0694C441D61604BC3ADE3DC945767A2ED56C0E167F220DD54194A610A6F	27-APR-06	5025

```

create or replace procedure criptare_card
as
    cheie_secreta raw(16);
    mod_operare pls_integer;
    cursor c_card is select id_card, numar, cvc, pin from card;
    v_numar_raw raw(100);
    v_cvc_raw raw(100);
    v_pin_raw raw(100);
    rezultat_numar raw(100);
    rezultat_cvc raw(100);
    rezultat_pin raw(100);

begin

    select cheie into cheie_secreta
    from tabel_chei_banca
    where UPPER(ume_tabel) = UPPER('CARD');

    dbms_output.put_line('Cheia de criptare pt tabelul CARD este ' || cheie_secreta);

```



```

mod_operare := dbms_crypto.encrypt_aes128 + dbms_crypto.pad_pkcs5 + dbms_crypto.chain_cbc;

for rec in c_card loop
    v_numar_raw := utl_i18n.string_to_raw(rec.numar, 'AL32UTF8');
    v_cvc_raw := utl_i18n.string_to_raw(rec.cvc, 'AL32UTF8');
    v_pin_raw := utl_i18n.string_to_raw(rec.pin, 'AL32UTF8');

    if length(v_numar_raw) < 64 then

        rezultat_numar := dbms_crypto.encrypt(v_numar_raw, mod_operare, cheie_secret);
        rezultat_cvc := dbms_crypto.encrypt(v_cvc_raw, mod_operare, cheie_secret);
        rezultat_pin := dbms_crypto.encrypt(v_pin_raw, mod_operare, cheie_secret);

        update card
        set numar = rezultat_numar, cvc = rezultat_cvc, pin = rezultat_pin
        where id_card = rec.id_card;

    end if;

end loop;

commit;
end;
/

begin
    criptare_card();
end;
/

```

Procedure CRIPTARE_CARD compiled

Cheia de criptare pt tabelul CARD este D997C091A03F5E5DA23FC12DDABC9545

PL/SQL procedure successfully completed.

ARD	ID_TIP_STATUS	ID_CONT	NUMAR	DATA_EMITERE	DATA_EXPIRARE	CVC	PIN
1	1	2	40 C4899C2FA0EC7322C459A82C633A58329641EAD9E7A5691AA091E518199EFBF2	21-AUG-05	21-AUG-05	BCF568033B3E7493168603E6DAB8B5B5	36B34EEB4397CE3B39E2EF0F897783F1
2	4	1	26 4064932D10A65EBF591FDE8DE0C9ABD5013BA343047EEBEE6136C885B4B8CDA3	18-MAY-19	18-MAY-19	457FF8C0335D015588C392AD74BF4A4C	94CDF9AB3D4B5BACCF1E31153553E6D2
3	4	1	5 BDF089DE24063D9565CFE0A221A99DBB040E7E6F6F39F00ECFB98CB3F7E4CEB	18-MAR-12	18-MAR-12	E49E32368FED6CFB37F532AA64E7B847	F39DAB03B6D2E32095D210C862FEFF1A
4	1	2	40 8DE3E5F8555AB7B4BBO69E8A88504A03875A0C4C9590C443B247068CF56926F	02-APR-98	02-APR-98	F192747C806A2FDDA008E7BA3D9F1E25	7AAA7E9A456F7A20560DFCDA0218B88
5	1	1	12 301391235DF55317E1C41FF400A174C5860CB7E45FC55E76551B48A9B40CBE82	24-SEP-09	24-SEP-09	EA2FF78F47B8EFF6E1B965307A8E2DE3	OCEAA47547B3CD0BE45C652A0B0D2F4A
6	4	2	4 8829F323B24152EE57A2BFE907B48E7C590B00FE7660F028A556A897AE8415E	15-JAN-21	15-JAN-21	5F48D74F55AACD558AA34A9ACA2C7BD0	72547AE89A248A37B67FAF3BC97432E6
7	4	2	16 B8D59154EE9873D204869636DEB98FE69C71204332A8F66927021C323A2A1D2	06-DEC-09	06-DEC-09	4A11547B14506C7382902244A3089CB3	2EDFFB1AD730F85A210EF31FE3578DCD
8	3	1	11 62B09C9F3F43214A4AA17D0DB6142ED9385A3A7809C8288AF1B81C29B4015A4	11-FEB-93	11-FEB-93	916161D6CDEF3754B4E3433D105D9BDB	33815041EF64A31C515CF8F4EC499E47
9	4	2	10 B0934D3BE5474BEFA651290093B96CBF22E16720FF9BE05397852EB0FA169B4	09-APR-16	09-APR-16	C8DCF8DAED3BC9FF7F1A7624BD5468B	F6271DA06062B84DBF202F521C07F6FA
10	3	1	11 6F284EEF5776B88FA741BB281529DD44818A78292147BB8DA06BBB70C420770A	04-MAR-06	04-MAR-06	0E80300B23A5564663A86GALL130ACA95	CB4A2F44E6C723B09D76CF3E9D5E0B48
11	3	1	37 B98F2D6787F2786ADA6DDFAD06F5F27C393C722DAC2E8A8C8210A6AF7EB4B652	26-FEB-94	26-FEB-94	8D380500AF5145FCE15F397812C82C2B	497FDF9DA8FF9A1A1C811436572FA9A7
12	4	2	13 7272F4C93C05A3BD05B3FC540C22B5F50C7E59E41A347B420725578FEC153FE5	21-OCT-03	21-OCT-03	AF24EDA091714AF72E95AE627D116F59	2043102D86D9E3799C1E99F60953469
13	2	2	7 457268D0A6A1B53CB0D40C7F18C53A4B9021E6AAFB4D0BFF80CEA387714FB57	18-APR-90	18-APR-90	768C397A8CCE77A5C00DF24B2D8364FA	18AC66FF9C588FB2276ADA226AF11C

3. Auditarea activităților asupra bazei de date

a. Auditare standard

Auditarea activității pe baza de date are două componente: monitorizarea și înregistrarea persistentă a unei mulțimi de activități și evenimente, stabilită a-priori, din baza de date. Prin auditarea standard audităm:

- comenzi SQL LDD, LMD și SELECT ce au loc asupra tabelelor dintr-o schemă din baza de date
- comenzi SQL pe un obiect specificat al bazei de date
- activitatea în rețea
- exercitarea privilegiilor
- sesiunea de lucru pe baza de date

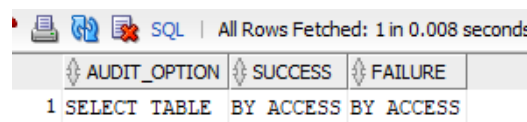
Pentru proiect, am ales să auditez primele două categorii, cu înregistrarea informațiilor monitorizate la nivelul bazei de date, în tabelul aud\$. Am exemplificat auditarea standard pe tabelele CONT și CARD din schema *user1*. Înainte de a activa auditările, am schimbat containerul cu cel de cdb\$root pentru a verifica valoarea parametrului audit_trail. Acesta avea valoarea XML, extended. Am modificat parametrul astfel încât informațiile monitorizate să fie înregistrate la nivelul bazei de date, după care am verificat auditul activ, folosind comenzile:

```
alter session set container=cdb$root;  
--datele despre auditari vor fi stocate la nivelul bazei de date  
alter system set audit_trail=db,extended scope=spfile;  
--verificam auditul activ  
select audit_option, success, failure from dba_stmt_audit_opts;
```

A trebuit să opresc instanța Oracle din SQL*PLUS cu comanda *shutdown immediate*; după care am repornit-o prin comanda *startup*; pentru ca modificarea valorii parametrului audit_trail să aibă efect.

În continuare, am modificat containerul cu cel în care am creat schema *user1* și am activat auditarea pe comanda SELECT.

```
--ne mutam pe conexiunea orclpdb si deschidem bd pluggable
alter session set container=orclpdb;
alter pluggable database orclpdb open;
audit select table;
select audit_option, success, failure from dba_stmt_audit_opts;
```



The screenshot shows the SQL Developer interface with the following table:

SQL All Rows Fetched: 1 in 0.008 seconds			
AUDIT_OPTION	SUCCESS	FAILURE	
1 SELECT TABLE	BY ACCESS	BY ACCESS	

Am rulat următoarele comenzi SELECT asupra tabelelor CONT și CARD, după care am afișat informațiile din tabelul aud\$.

```
--am rulat comenzile ca utilizator user1
select * from cont;
select * from card;
select * from cont
where data_emitere > sysdate - 30 and data_emitere < sysdate;
select * from cont
where data_creatie > sysdate - 30 and data_creatie < sysdate;
select count(*)
from cont
where sold > 7000;
```

```
--sa se afiseze comenzile SELECT efectuate asupra tabelelor CARD si CONT
select obj$creator, obj$name, userid, sqltext, ntimestamp#
from aud$
where upper(obj$name) in ('CARD', 'CONT')
order by obj$name;
```

	OBJ\$CREATOR	OBJ\$NAME	USERID	SQLTEXT	NTIMESTAMP#
1	USER1	CARD	USER1	select * from card	22-JAN-23 11.13.06.5830000
2	USER1	CONT	USER1	select * from cont	22-JAN-23 11.12.56.7680000
3	USER1	CONT	USER1	select * from contwhere data_emitere > sysdate - 30 and data_emitere < s...	22-JAN-23 11.18.34.1000000
4	USER1	CONT	USER1	select * from contwhere data_emitere > sysdate - 30 and data_emitere < s...	22-JAN-23 11.18.34.1580000
5	USER1	CONT	USER1	select * from contwhere data_creatie > sysdate - 30 and data_creatie < sys...	22-JAN-23 11.18.50.8230000
6	USER1	CONT	USER1	select count(*)from contwhere sold > 7000	22-JAN-23 11.19.47.1410000

Am realizat și un raport prin care se afișează numărul de comenzi SELECT efectuate asupra tabelelor CONT și CARD:

- de fiecare utilizator, pentru fiecare tabel
- de fiecare utilizator, indiferent de tabel
- indiferent de utilizator și tabel

```
select userid, obj$creator, obj$name, count(*)
from aud$
where upper(obj$name) in ('CONT','CARD')
group by rollup (userid, (obj$creator, obj$name));
```

SQL All Rows Fetched: 4 in 0.005 seconds				
	USERID	OBJ\$CREATOR	OBJ\$NAME	COUNT(*)
1	USER1	USER1	CARD	1
2	USER1	USER1	CONT	5
3	USER1	(null)	(null)	6
4	(null)	(null)	(null)	6

De asemenea, am activat și auditarea asupra comenzilor LMD INSERT, UPDATE, DELETE pe tabelul CONT doar atunci când comanda eșuează.

```
audit insert, update, delete on user1.cont whenever not successful;
select object_name, object_type, owner, sel, ins, upd, del from dba_obj_audit_opts
where lower(object_name) = 'cont';
```

OBJECT_NAME	OBJECT_TYPE	OWNER	SEL	INS	UPD	DEL
1 CONT	TABLE	USER1	-/-	-/A	-/A	-/A

În schema user1, am rulat comanda următoare, care va produce o eroare generată de constrângerea de cheie primară (mai există o înregistrare cu cheia primară 1):

```
insert into CONT (id_cont, id_tip_cont, id_tip_status, IBAN, data_creat, sold) values (1, 1, 1, 'SN26732036098651', '12-Sep-2018', 299);
```

```
select obj$creator, obj$name, userid, sqltext, ntimestamp#
from aud$
where upper(obj$name) in ('CONT');
```

OBJ\$CREATOR	OBJ\$NAME	USERID	SQLTEXT	NTIMESTAMP#
1 USER1	CONT	USER1	select * from cont	22-JAN-23 11.12.56.768000000 AM
2 USER1	CONT	USER1	select * from contwhere data_emitere > sysdate - 30 and data_emitere < s...	22-JAN-23 11.18.34.100000000 AM
3 USER1	CONT	USER1	select * from contwhere data_emitere > sysdate - 30 and data_emitere < s...	22-JAN-23 11.18.34.158000000 AM
4 USER1	CONT	USER1	select * from contwhere data_creat > sysdate - 30 and data_creat < sys...	22-JAN-23 11.18.50.823000000 AM
5 USER1	CONT	USER1	select count(*)from contwhere sold > 7000	22-JAN-23 11.19.47.141000000 AM
6 USER1	CONT	USER1	insert into CONT (id_cont, id_tip_cont, id_tip_status, IBAN, data_creat...	22-JAN-23 11.33.34.209000000 AM

În final, am oprit toate auditările pentru comenzile SQL nelegate de un obiect specific.

```
noaudit all;
```

b. Trigger-i de auditare

Pentru auditare, am creat trigger-i personalizați care să înregistreze anumite informații de interes. În cazul de față, am ales să monitorizez informații despre operațiunile LMD asupra tabelului CLIENT. Informațiile monitorizate vor fi stocate într-un tabel special creat, TAB_AUDIT_CLIENT.

```
create table tab_audit_client (
  id_secv number(4) primary key,
  user_ varchar2(20),
  session_ number(10),
```



```

create or replace trigger audit_client_after
after insert on user1.client
declare
    nr_inreg_after number;
    nr_inreg_before number;
    current_session varchar2(100);
    id_rec_audit number;
begin
    select count(*) into nr_inreg_after from user1.client;
    dbms_output.put_line('Nr inregistrari dupa inserare: ' || nr_inreg_after);
    select sys_context('userenv', 'sessionid') into current_session from dual;
    select max(id_secv) into id_rec_audit from tab_audit_client where session_ =
current_session;
    select delta_records into nr_inreg_before from tab_audit_client
    where id_secv = id_rec_audit;
    update tab_audit_client
    set delta_records = nr_inreg_after
    where id_secv = id_rec_audit;

end;
/

```

Am testat cei doi triggeri prin rularea unei comenzi de inserare pe tabela CLIENT:

```

insert into CLIENT (id_client, id_banca, nume, prenume, data_nastere, CNP) values (100, 40,
'James', 'Cohen', '20-Sep-1999', '1622949279576');
commit;

```

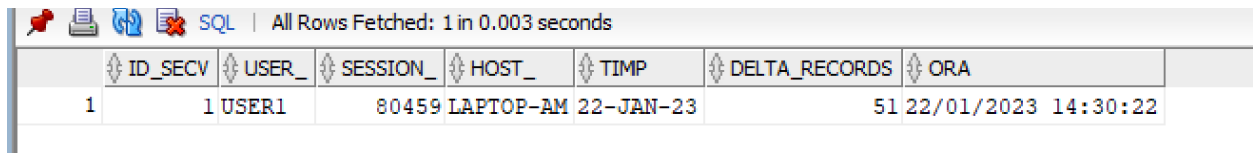
Am verificat în tabelul de audit ce date au fost inserate:

```

select t.*, to_char(timp, 'dd/mm/yyyy hh24:mi:ss') as Ora

```

```
from sys.tab_audit_client t;
```



ID_SECV	USER_	SESSION_	HOST_	TIMP	DELTA_RECORDS	ORA
1	1	USER1	80459	LAPTOP-AM	22-JAN-23	51 22/01/2023 14:30:22

ii. Am creat un trigger care înregistrează în același tabel TAB_AUDIT_CLIENT:

- valoarea 1 atunci când este modificat un client cu data de naștere după anul 2000
- valoarea -1 atunci când este șters un client cu data de naștere după anul 2000

```
create or replace trigger data_client
after delete or update on user1.client
for each row
begin
    if UPDATING and :new.data_nastere > '01-JAN-2000' then
        insert into tab_audit_client values (secv_audit_client.nextval, sys_context('userenv',
'session_user'),
            sys_context('userenv', 'sessionid'), sys_context('userenv', 'host'), sysdate, 1);
    elsif DELETING and :old.data_nastere > '01-JAN-2000' then
        insert into tab_audit_client values (secv_audit_client.nextval, sys_context('userenv',
'session_user'),
            sys_context('userenv', 'sessionid'), sys_context('userenv', 'host'), sysdate, -1);
    end if;
end;
```

S-au executat următoarele comenzi care au declanșat triggerul data_client:

```
insert into CLIENT (id_client, id_banca, nume, prenume, data_nastere, CNP) values (101, 40,
'Maria', 'Olteanu', '20-Sep-2000', '1622949279576');
commit;
```



```

update client
set id_banca = 10
where id_client = 101;
commit;

delete from client
where id_client = 101;
commit;
drop trigger data_client;

```

Am verificat din nou tabelul de audit, obținând rezultatul:

	ID_SECV	USER_	SESSION_	HOST_	TIMP	DELTA_RECORDS	ORA
1	2	USER1	80459	LAPTOP-AM	22-JAN-23	51	22/01/2023 14:39:02
2	3	USER1	80459	LAPTOP-AM	22-JAN-23	52	22/01/2023 14:39:21
3	4	USER1	80459	LAPTOP-AM	22-JAN-23	1	22/01/2023 14:39:23
4	6	USER1	80459	LAPTOP-AM	22-JAN-23	-1	22/01/2023 14:41:51

c. Politici de auditare

Am definit o politică de auditare prin care se înregistrează instrucțiunile LMD de modificare a soldului unui cont pe tabelul CONT.

```

create or replace procedure proc_audit_alert (object_schema varchar2, object_name varchar2,
                                             policy_name varchar2)
as
begin
    dbms_output.put_line('Inercare modificare sold cont');
end;
/

```

```
create or replace procedure proc_audit_sold as
begin
  dbms_fga.add_policy(
    object_schema => 'USER1',
    object_name => 'CONT',
    policy_name => 'policy_sold_cont',
    enable => false,
    statement_types => 'UPDATE',
    handler_module => 'PROC_AUDIT_ALERT');
end;
/
```

Am executat procedura prin care s-a adăugat politica definită.

```
execute proc_audit_sold;
```

Am acitvat politica de auditare:

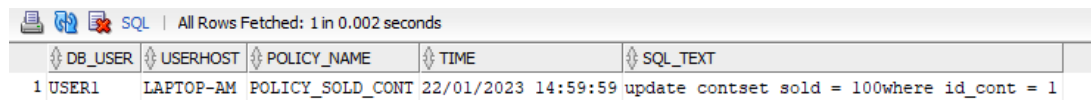
```
begin
  dbms_fga.enable_policy(object_schema => 'USER1',
    object_name => 'CONT',
    policy_name => 'policy_sold_cont');
end;
/
```

Am rulat comand de UPDATE asupra coloanei cont:

```
update cont
set sold = 100
where id_cont = 1;
```

Am verificat vizualizarea DBA_FGA_AUDIT_TRAIL pentru rezultatele auditului.

```
select db_user, userhost, policy_name, to_char(timestamp, 'dd/mm/yyyy hh24:mi:ss') Time,  
sql_text  
from dba_fga_audit_trail  
order by timestamp desc;
```



DB_USER	USERHOST	POLICY_NAME	TIME	SQL_TEXT
1 USER1	LAPTOP-AM	POLICY_SOLD_CONT	22/01/2023 14:59:59	update contset sold = 100where id_cont = 1

În final, am dezactivat politica de auditare.

```
begin  
  dbms_fga.disable_policy(object_schema => 'USER1',  
    object_name => 'CONT',  
    policy_name => 'policy_sold_cont');  
end;  
/
```

4. Gestiunea utilizatorilor unei baze de date și a resurselor computaționale

a. Proiectarea configurației de management a identităților în baza de date (matricile proces-utilizator, entitate-proces, entitate-utilizator)

Parcurgerea etapelor de proiectare în contextul unei aplicații bancare:

1) Utilizatorii aplicației sunt:

- Reprezentanții serviciului pentru clienți – CNP
- Clienții înregistrați ce și-au deschis conturi la banca respectivă – CNP
- Personalul de marketing și vânzări – CNP
- Personalul de detectare a fraudelor – CNP
- Administratorul aplicației și al bazei de date – CNP

Un alt atribut de identificare mai comun îl reprezintă adresa de e-mail confirmată. Însă, în mod uzual se construiește un tandem *atribut de identificare – parola*.

2) În cadrul aplicației există următoarele procese

P1: Introducerea unei sucursale noi

P2: Introducerea unui client nou

P3: Adăugarea unui cont asociat unui client

P4: Adăugarea unui card asociat unui cont

P5: Vizualizarea conturilor pe care le-a deschis un client

P6: Vizualizarea cardurilor emise pentru un anumit cont

P7: Închiderea unui cont

P8: Închiderea/Blocarea unui card

P9: Interogare sold pentru contul unui client

P10: Vizualizarea sucursalelor băncii pentru un anumit oraș și țară

P11: Modificarea soldului unui cont

P12: Modificarea codului PIN al unui card

P13: Vizualizarea clienților pentru o anumită sucursală

P14: Vizualizarea tipurilor de conturi

P15: Adăugarea unui nou tip de cont

P16: Vizualizarea conturilor active pentru o sucursală

P17: Vizualizarea cardurilor în funcție de tipul cardului pentru o sucursală

5) Construirea matricii entitate-utilizator

	Reprezentanții serviciului pentru clienți	Clienții	Personalul de marketing și vânzări	Personalul de detectare a fraudelor	Administratorul aplicației și al bazei de date
TARA	S	S	S	S	S
ORAS	S	S	S	S	S
BANCA	S	S	S	S	I,U,S
CLIENT	I,U,S	S	S	S	S
CLIENT_CONT	I,U,S	S	S	S	
CONT	S,I,D	S,U,D	S	S	S
TIP_CONT	S	S	S	S	I,U
CARD	I,U,D	S,U,D	S	S	S
TIP_CARD	S	S	S	S	S
STATUS	S	S	S	S	S

b. Implementarea configurației de management a identităților în baza de date

Pentru implementarea configurației de management a identităților sunt necesare mai multe etape, care vor fi efectuate de către administratorul bazei de date.

I. Crearea conturilor utilizatorilor

Utilizatorii vor fi creați cu clauza de parolă expirată, pentru a-și putea modifica parola când se va conecta pentru prima oară. Este recomandată această practică pentru a nu exista suspiciunea că parola este cunoscută și de administratorul bazei de date. Conturile de utilizatori au fost create pe baza punctului 4.a.1.

```
--creare utilizatori
--utilizator admin

create user bank_admin identified by parolaadmin password expire;
grant create session to bank_admin;

--utilizator reprezentanti serviciu pentru clienti
create user bank_consultant1 identified by parolaconstultant password expire;
```

```

grant create session to bank_consultant1;

create user bank_consultant2 identified by parolaconstultant password expire;
grant create session to bank_consultant2;

--utilizatori clienti
create user bank_client1 identified by parolaclient password expire;
grant create session to bank_client1;

create user bank_client2 identified by parolaclient password expire;
grant create session to bank_client2;

create user bank_client3 identified by parolaclient password expire;
grant create session to bank_client3;

--utilizator personal de marketing
create user bank_marketing identified by parolamarketing password expire;
grant create session to bank_marketing;

--utilizator personal de detectare a fraudelor
create user bank_frauda identified by parolafrauda password expire;
grant create session to bank_frauda;

select username, account_status, expiry_date
from dba_users where lower(username) like '%bank%';

```

SQL All Rows Fetched: 8 in 0.005 seconds			
	USERNAME	ACCOUNT_STATUS	EXPIRY_DATE
1	BANK_CLIENT2	EXPIRED	24-JAN-23
2	BANK_CONSULTANT2	EXPIRED	24-JAN-23
3	BANK_CONSULTANT1	EXPIRED	24-JAN-23
4	BANK_CLIENT3	EXPIRED	24-JAN-23
5	BANK_FRAUDA	EXPIRED	24-JAN-23
6	BANK_CLIENT1	EXPIRED	24-JAN-23
7	BANK_MARKETING	EXPIRED	24-JAN-23
8	BANK_ADMIN	EXPIRED	24-JAN-23

II. Stabilirea aspectelor referitoare la spații de stocare pentru utilizatorii creați

Dimensiunea va fi împărțită în cote astfel:

- Utilizatorul BANK_ADMIN: nelimitat
- Utilizatorii BANK_CONSULTANT1 și BANK_CONSULTANT2: câte 3MB fiecare

- Utilizatorul BANK_FRAUDA: 1MB
- Restul utilizatorilor: 0MB

--alocarea cotelor de spatiu de stocare al tablespace-ului USERS pentru toti utilizatorii

```
alter user bank_admin quota unlimited on users;
alter user bank_consultant1 quota 3M on users;
alter user bank_consultant2 quota 3M on users;
alter user bank_client1 quota 0M on users;
alter user bank_client2 quota 0M on users;
alter user bank_client3 quota 0M on users;
alter user bank_marketing quota 0M on users;
alter user bank_frauda quota 1M on users;
```

```
desc dba_ts_quotas;
select * from dba_ts_quotas
where username like '%BANK%';
```

SQL | All Rows Fetched: 4 in 0.014 seconds

	TABLESPACE_NAME	USERNAME	BYTES	MAX_BYTES	BLOCKS	MAX_BLOCKS	DROPPED
1	USERS	BANK_ADMIN	0	-1	0	-1	NO
2	USERS	BANK_CONSULTANT2	0	3145728	0	384	NO
3	USERS	BANK_FRAUDA	0	1048576	0	128	NO
4	USERS	BANK_CONSULTANT1	0	3145728	0	384	NO

III. Stabilirea accesului la resurse computaționale și alte configurări

Am grupat restricțiile de acces la resurse în profiluri și am atașat profilurile utilizatorilor:

- pentru utilizatorii reprezentanți ai serviciului pentru clienți (consultanții) și clienți, consumul de CPU per apel să nu depășească 60 de secunde, să aibă dreptul la o singură sesiune la un moment dat, timpul de viață al parolei să fie de 20 zile și să poată greși parola de maxim 5 ori
- pentru utilizatorii din personalul de marketing și vânzări, numărul maxim de conexiuni permise să fie 2, timpul maxim de inactivitate permis să fie de 2 minute, iar timpul total de conectare să nu depășească 10 minute chiar în perioadele de activitate

- pentru utilizatorii din personalul de detectare a fraudelor, numărul maxim de conexiuni permise să fie 2, timpul maxim de inactivitate permis să fie de 3 minute, timpul de viață al parolei să fie de 14 zile și să poată greși parola de maxim 3 ori

```
--pentru utilizatorii reprezentanti ai serviciului pentru clienti si clienti, consumul de CPU per  
apel sa nu depaseasca 60 de secunde,  
--sa aiba dreptul la o singura sesiune la un moment dat, timpul de viata al parolei sa fie de 20  
zile si sa poata gresi parola de maxim 5 ori
```

```
create profile bank_profil_consultant_client limit  
  cpu_per_call 6000  
  sessions_per_user 1  
  password_life_time 20  
  failed_login_attempts 5;
```

```
--pentru utilizatorii din personalul de marketing si vanzari, numarul maxim de conexiuni  
permise sa fie 2,  
--timpul maxim de inactivitate permis sa fie de 2 minute, iar timpul total de conectare sa nu  
depaseasca 10 minute chiar in perioadele de activitate
```

```
create profile bank_profil_marketing limit  
  sessions_per_user 2  
  idle_time 2  
  connect_time 10;
```

```
--pentru utilizatorii din personalul de detectare a fraudelor, numarul maxim de conexiuni  
permise sa fie 2,  
--timpul maxim de inactivitate permis sa fie de 3 minute, timpul de viata al parolei sa fie de 14  
zile si sa poata gresi parola de maxim 3 ori
```

```
create profile bank_profil_frauda limit  
  sessions_per_user 2  
  idle_time 3  
  password_life_time 20  
  failed_login_attempts 3;
```

SQL | All Rows Fetched: 51 in 0.014 seconds

	PROFILE	RESOURCE_NAME	RESOURCE_TYPE	LIMIT	COMMON	INHERITED	IMPLICIT
1	BANK_PROFIL_CONSULTANT_CLIENT	COMPOSITE_LIMIT	KERNEL	DEFAULT	NO	NO	NO
2	BANK_PROFIL_FRAUDA	COMPOSITE_LIMIT	KERNEL	DEFAULT	NO	NO	NO
3	BANK_PROFIL_MARKETING	COMPOSITE_LIMIT	KERNEL	DEFAULT	NO	NO	NO
4	BANK_PROFIL_CONSULTANT_CLIENT	SESSIONS_PER_USER	KERNEL	1	NO	NO	NO
5	BANK_PROFIL_FRAUDA	SESSIONS_PER_USER	KERNEL	2	NO	NO	NO
6	BANK_PROFIL_MARKETING	SESSIONS_PER_USER	KERNEL	2	NO	NO	NO
7	BANK_PROFIL_CONSULTANT_CLIENT	CPU_PER_SESSION	KERNEL	DEFAULT	NO	NO	NO
8	BANK_PROFIL_FRAUDA	CPU_PER_SESSION	KERNEL	DEFAULT	NO	NO	NO
9	BANK_PROFIL_MARKETING	CPU_PER_SESSION	KERNEL	DEFAULT	NO	NO	NO
10	BANK_PROFIL_CONSULTANT_CLIENT	CPU_PER_CALL	KERNEL	6000	NO	NO	NO
11	BANK_PROFIL_FRAUDA	CPU_PER_CALL	KERNEL	DEFAULT	NO	NO	NO
12	BANK_PROFIL_MARKETING	CPU_PER_CALL	KERNEL	DEFAULT	NO	NO	NO
13	BANK_PROFIL_CONSULTANT_CLIENT	LOGICAL_READS_PER_SESSION	KERNEL	DEFAULT	NO	NO	NO
14	BANK_PROFIL_FRAUDA	LOGICAL_READS_PER_SESSION	KERNEL	DEFAULT	NO	NO	NO
15	BANK_PROFIL_MARKETING	LOGICAL_READS_PER_SESSION	KERNEL	DEFAULT	NO	NO	NO
16	BANK_PROFIL_CONSULTANT_CLIENT	LOGICAL_READS_PER_CALL	KERNEL	DEFAULT	NO	NO	NO
17	BANK_PROFIL_FRAUDA	LOGICAL_READS_PER_CALL	KERNEL	DEFAULT	NO	NO	NO

Am atribuit utilizatorilor creați profilurile definite anterior.


```
--atribuire profiluri pentru useri
alter user bank_consultant1 profile bank_profil_consultant_client;
alter user bank_consultant2 profile bank_profil_consultant_client;
alter user bank_client1 profile bank_profil_consultant_client;
alter user bank_client2 profile bank_profil_consultant_client;
alter user bank_client3 profile bank_profil_consultant_client;
alter user bank_marketing profile bank_profil_marketing;
alter user bank_frauda profile bank_profil_frauda;
```

Utilizatorul bank_consultant1 și-a schimbat parola și s-a putut conecta la o conexiune nouă cu username și parola.

	USERNAME	ACCOUNT_STATUS	EXPIRY_DATE
1	BANK_ADMIN	EXPIRED	24-JAN-23
2	BANK_CLIENT2	EXPIRED	24-JAN-23
3	BANK_CONSULTANT2	EXPIRED	24-JAN-23
4	BANK_CONSULTANT1	OPEN	13-FEB-23
5	BANK_CLIENT3	EXPIRED	24-JAN-23
6	BANK_CLIENT1	EXPIRED	24-JAN-23
7	BANK_FRAUDA	EXPIRED	24-JAN-23
8	BANK_MARKETING	EXPIRED	24-JAN-23

New / Select Database Connection

Connection Name	Connection Details
dw_project	test2@//localhos...
dw_project_oltp	oltp@//localhost:...
hr_local	hr@//localhost:1...
local_db_o19c	sys as sysdba@/...
master_dw_if	master_dw_if@/...
sec_bd_project	user1@//localhos...
securitate_bd	secbd30@//193....
test2_local	test2@//localhos...

Name:  Color

Database Type:

User Info Proxy User

Authentication Type:

Username: Role:

Password: ☐ Save Password

Connection Type:

Details Advanced

Hostname:

Port:

☐ SID


☒ Service name

Status : Success

Utilizatorul bank_consultant1 a greșit parola de 5 ori, în consecință contul i-a fost blocat.

New / Select Database Connection

Connection Name	Connection Details
dw_project	test2@//localhos...
dw_project_oltp	oltp@//localhost:...
hr_local	hr@//localhost:1...
local_db_o19c	sys as sysdba@/...
master_dw_if	master_dw_if@/...
sec_bd_project	user1@//localhos...
securitate_bd	secbd30@//193....
test2_local	test2@//localhos...

Name:  Color

Database Type:

User Info Proxy User

Authentication Type:

Username: Role:

Password: ☐ Save Password

Connection Type:

Details Advanced

Hostname:

Port:

☐ SID

☒ Service name

Status : Failure -Test failed: ORA-28000: The account is locked.

	USERNAME	ACCOUNT_ST...	EXPIRY_DATE
1	BANK_ADMIN	EXPIRED	24-JAN-23
2	BANK_CLIENT2	EXPIRED	24-JAN-23
3	BANK_CONSULTANT2	EXPIRED	24-JAN-23
4	BANK_CONSULTANT1	LOCKED (TIMED)	13-FEB-23
5	BANK_CLIENT3	EXPIRED	24-JAN-23
6	BANK_CLIENT1	EXPIRED	24-JAN-23
7	BANK_FRAUDA	EXPIRED	24-JAN-23
8	BANK_MARKETING	EXPIRED	24-JAN-23

Toți utilizatorii și-au schimbat parolele și le-au fost deblocate conturile.

SQL All Rows Fetched: 8 in 0.003 seconds			
	USERNAME	ACCOUNT_STATUS	EXPIRY_DATE
1	BANK_ADMIN	OPEN	23-JUL-23
2	BANK_CLIENT2	OPEN	13-FEB-23
3	BANK_CONSULTANT2	OPEN	13-FEB-23
4	BANK_CONSULTANT1	OPEN	13-FEB-23
5	BANK_CLIENT3	OPEN	13-FEB-23
6	BANK_CLIENT1	OPEN	13-FEB-23
7	BANK_FRAUDA	OPEN	13-FEB-23
8	BANK_MARKETING	OPEN	23-JUL-23

IV. Configurarea unui plan de resurse cu următoarele reguli:

Vor exista 4 grupuri de consum:

- management(bank_admin) – consum de CPU 20%
- consultanta(bank_consultant) – consum de CPU 30%
- clienți(bank_client) – consum de CPU 30%
- other_departments (bank_marketing, bank_frauda) – consum de CPU 10%
- other_groups – consum de CPU 10%

```
CREATE OR REPLACE PROCEDURE bank_plan_consum AS
  N NUMBER :=0;
BEGIN
  DBMS_RESOURCE_MANAGER.CREATE_PENDING_AREA();
  DBMS_RESOURCE_MANAGER.CREATE_PLAN(PLAN => 'BANK_plan1',
    COMMENT => 'Acesta este un plan pentru sistemul de banking');
  --grupuri de consum
```

```
DBMS_RESOURCE_MANAGER.CREATE_CONSUMER_GROUP(CONSUMER_GROUP
=> 'management', COMMENT => 'Acesta grupeaza sesiunile utilizatorilor care administreaza
aplicatia');
```

```
DBMS_RESOURCE_MANAGER.CREATE_CONSUMER_GROUP(CONSUMER_GROUP
=> 'consultanta', COMMENT => 'Acesta grupeaza sesiunile utilizatorilor care ofera
consultanta');
```

```
DBMS_RESOURCE_MANAGER.CREATE_CONSUMER_GROUP(CONSUMER_GROUP
=> 'clienti', COMMENT => 'Acesta grupeaza sesiunile utilizatorilor care primesc
consultanta');
```

```
DBMS_RESOURCE_MANAGER.CREATE_CONSUMER_GROUP(CONSUMER_GROUP
=> 'other_departments', COMMENT => 'Acesta grupeaza sesiunile utilizatorilor din
departamentele de marketing si detectare fraude');
```

```
--se va crea doar daca nu exista deja
SELECT COUNT(*) INTO n
FROM DBA_RSRC_CONSUMER_GROUPS
WHERE CONSUMER_GROUP='OTHER_GROUPS';
```

```
IF n=0 THEN
```

```
DBMS_RESOURCE_MANAGER.CREATE_CONSUMER_GROUP(CONSUMER_GROUP
=> 'OTHER_GROUPS', COMMENT => 'Acesta grupeaza RESTUL LUMII');
END IF;
```

```
--mapari statice utilizatori pe grupuri consum, nu pot fi mapati pe grupul OTHER_GROUPS
```

```
DBMS_RESOURCE_MANAGER.SET_CONSUMER_GROUP_MAPPING(DBMS_RESOU
RCE_MANAGER.ORACLE_USER, 'bank_admin', 'management');
```

```
DBMS_RESOURCE_MANAGER.SET_CONSUMER_GROUP_MAPPING(DBMS_RESOU
RCE_MANAGER.ORACLE_USER, 'bank_consultant1', 'consultanta');
```

```
DBMS_RESOURCE_MANAGER.SET_CONSUMER_GROUP_MAPPING(DBMS_RESOU
RCE_MANAGER.ORACLE_USER, 'bank_consultant2', 'consultanta');
```

```
DBMS_RESOURCE_MANAGER.SET_CONSUMER_GROUP_MAPPING(DBMS_RESOU
RCE_MANAGER.ORACLE_USER, 'bank_client1', 'clienti');
```

```
DBMS_RESOURCE_MANAGER.SET_CONSUMER_GROUP_MAPPING(DBMS_RESOU
RCE_MANAGER.ORACLE_USER, 'bank_client2', 'clienti');
```

```
DBMS_RESOURCE_MANAGER.SET_CONSUMER_GROUP_MAPPING(DBMS_RESOURCE_MANAGER.OracleUser, 'bank_client3', 'clienti');
```

```
DBMS_RESOURCE_MANAGER.SET_CONSUMER_GROUP_MAPPING(DBMS_RESOURCE_MANAGER.OracleUser, 'bank_marketing', 'other_departments');
```

```
DBMS_RESOURCE_MANAGER.SET_CONSUMER_GROUP_MAPPING(DBMS_RESOURCE_MANAGER.OracleUser, 'bank_frauda', 'other_departments');
```

```
--directivile de plan pentru fiecare grup de consum
DBMS_RESOURCE_MANAGER.CREATE_PLAN_DIRECTIVE(PLAN =>
'BANK_plan1', GROUP_OR_SUBPLAN => 'management',
      COMMENT => 'directiva de plan pt gr management', MGMT_P1 => 20);
DBMS_RESOURCE_MANAGER.CREATE_PLAN_DIRECTIVE(PLAN =>
'BANK_plan1', GROUP_OR_SUBPLAN => 'consultanta',
      COMMENT => 'directiva de BANK_plan1 pt gr consultanta', MGMT_P1
=> 30);
DBMS_RESOURCE_MANAGER.CREATE_PLAN_DIRECTIVE(PLAN =>
'BANK_plan1', GROUP_OR_SUBPLAN => 'clienti',
      COMMENT => 'directiva de plan pt gr clienti', MGMT_P1 => 30);
DBMS_RESOURCE_MANAGER.CREATE_PLAN_DIRECTIVE(PLAN =>
'BANK_plan1', GROUP_OR_SUBPLAN => 'other_departments',
      COMMENT => 'directiva de plan pt gr departamentul de marketing si
frauda', MGMT_P1 => 10);
DBMS_RESOURCE_MANAGER.CREATE_PLAN_DIRECTIVE(PLAN =>
'BANK_plan1', GROUP_OR_SUBPLAN => 'OTHER_GROUPS',
      COMMENT => 'directiva de plan pt gr alte grupuri', MGMT_P1 => 10);

DBMS_RESOURCE_MANAGER.VALIDATE_PENDING_AREA();
DBMS_RESOURCE_MANAGER.SUBMIT_PENDING_AREA();

END;
/
```

```
execute bank_plan_consum;
```

```
desc dba_rsrc_consumer_groups;
select * from dba_rsrc_consumer_groups
order by consumer_group;
```

	USERNAME	INITIAL_RSRC_CONSUMER_GROUP
1	BANK_CONSULTANT2	CONSULTANTA
2	BANK_CLIENT1	CLIENTI
3	BANK_CLIENT2	CLIENTI
4	BANK_CLIENT3	CLIENTI
5	BANK_MARKETING	OTHER_DEPARTMENTS
6	BANK_FRAUDA	OTHER_DEPARTMENTS
7	BANK_ADMIN	MANAGEMENT
8	BANK_CONSULTANT1	CONSULTANTA

```
select distinct u.username, d.group_or_subplan, d.mgmt_p1, plan
from dba_rsrc_plan_directives d
left join dba_users u on (d.group_or_subplan = u.INITIAL_RSRC_CONSUMER_GROUP)
order by u.username nulls last;
```

FETCHED 30 ROWS IN 0.044 SECONDS			
	USERNAME	GROUP_OR_SUBPLAN	MGMT_P1 PLAN
1	BANK_ADMIN	MANAGEMENT	20 BANK_PLAN1
2	BANK_CLIENT1	CLIENTI	30 BANK_PLAN1
3	BANK_CLIENT2	CLIENTI	30 BANK_PLAN1
4	BANK_CLIENT3	CLIENTI	30 BANK_PLAN1
5	BANK_CONSULTANT1	CONSULTANTA	30 BANK_PLAN1
6	BANK_CONSULTANT2	CONSULTANTA	30 BANK_PLAN1
7	BANK_FRAUDA	OTHER_DEPARTMENTS	10 BANK_PLAN1
8	BANK_MARKETING	OTHER_DEPARTMENTS	10 BANK_PLAN1

5. Privilegii si roluri

a. Privilegii sistem si obiect

I. Privilegii sistem

- Conectat ca SYS AS SYSDBA, ofer privilegiile utilizatorului BANK_ADMIN să se conecteze la baza de date și să creeze orice obiect de un anumit tip și aparține propriei scheme sau unei scheme a unui alt utilizator

```
GRANT CREATE SESSION TO bank_admin;  
GRANT CREATE TABLE TO bank_admin;  
GRANT CREATE ANY TABLE TO bank_admin;  
GRANT CREATE ANY INDEX TO bank_admin;
```

	OWNER	OBJECT_NAME
1	BANK_ADMIN	BANCA
2	BANK_ADMIN	BANCA_PK
3	BANK_ADMIN	CARD
4	BANK_ADMIN	CARD_PK
5	BANK_ADMIN	CLIENT
6	BANK_ADMIN	CLIENT_PK
7	BANK_ADMIN	CLIENT_CONT
8	BANK_ADMIN	CLIENT_CONT_PK
9	BANK_ADMIN	CONT
10	BANK_ADMIN	CONT_PK
11	BANK_ADMIN	ORAS
12	BANK_ADMIN	ORAS_PK
13	BANK_ADMIN	STATUS
14	BANK_ADMIN	STATUS_PK
15	BANK_ADMIN	TARA
16	BANK_ADMIN	TARA_PK
17	BANK_ADMIN	TIP_CARD
18	BANK_ADMIN	TIP_CARD_PK
19	BANK_ADMIN	TIP_CONT
20	BANK_ADMIN	TIP_CONT_PK

II. Privilegii asupra obiectelor schemei

Pe baza matricii entitate-utilizator se vor acorda următoarele privilegii asupra obiectelor unei scheme:

- Administratorul bazei de date (a băncii) are dreptul de a insera date în tabelele TARA, ORAS, BANCA, TIP_CONT, TIP_CARD, STATUS din propria schemă.
- Consultanții băncii BANK_CONSULTANT1 ȘI BANK_CONSULTANT2 vor avea acces la toate tabelele din schema adminului pentru a obține informațiile necesare despre un client al băncii. În plus, vor primi dreptul de a insera, actualiza și șterge date despre client, conturi și carduri

```
GRANT SELECT ON bank_admin.tara to bank_consultant1, bank_consultant2;
GRANT SELECT ON bank_admin.oras to bank_consultant1, bank_consultant2;
GRANT SELECT ON bank_admin.banca to bank_consultant1, bank_consultant2;
GRANT SELECT ON bank_admin.client to bank_consultant1, bank_consultant2;
GRANT SELECT ON bank_admin.client_cont to bank_consultant1, bank_consultant2;
GRANT SELECT ON bank_admin.cont to bank_consultant1, bank_consultant2;
GRANT SELECT ON bank_admin.tip_cont to bank_consultant1, bank_consultant2;
GRANT SELECT ON bank_admin.card to bank_consultant1, bank_consultant2;
GRANT SELECT ON bank_admin.tip_card to bank_consultant1, bank_consultant2;
GRANT SELECT ON bank_admin.status to bank_consultant1, bank_consultant2;

GRANT INSERT, UPDATE, DELETE ON bank_admin.client to bank_consultant1,
bank_consultant2;
GRANT INSERT, UPDATE, DELETE ON bank_admin.cont to bank_consultant1,
bank_consultant2;
GRANT INSERT, UPDATE, DELETE ON bank_admin.client_cont to bank_consultant1,
bank_consultant2;
GRANT INSERT, UPDATE, DELETE ON bank_admin.card to bank_consultant1,
bank_consultant2;
```

- Clienții vor primi acces de vizualizare a tabelelor din schema administratorului și va primi dreptul de a-și modifica codul pin de la card. Pentru ultimul privilegiu, se va crea un rol corespunzător.

```
GRANT SELECT ON bank_admin.tip_cont to bank_client1, bank_client2, bank_client3;  
GRANT SELECT ON bank_admin.tip_card to bank_client1, bank_client2, bank_client3;  
GRANT SELECT ON bank_admin.status to bank_client1, bank_client2, bank_client3;  
GRANT SELECT ON bank_admin.banca to bank_client1, bank_client2, bank_client3;  
GRANT SELECT ON bank_admin.oras to bank_client1, bank_client2, bank_client3;  
GRANT SELECT ON bank_admin.tara to bank_client1, bank_client2, bank_client3;  
GRANT SELECT ON bank_admin.card to bank_client1, bank_client2, bank_client3;  
GRANT SELECT ON bank_admin.cont to bank_client1, bank_client2, bank_client3;  
GRANT SELECT ON bank_admin.client_cont to bank_client1, bank_client2, bank_client3;  
GRANT SELECT ON bank_admin.client to bank_client1, bank_client2, bank_client3;
```

```
create role update_pin_card;  
grant update(pin) on bank_admin.card to update_pin_card;  
grant update_pin_card to bank_client1;  
grant update_pin_card to bank_client2;  
grant update_pin_card to bank_client3;
```

```
update bank_admin.card  
set id_tip_card = 2  
where id_card = 1;  
  
update bank_admin.card  
set pin = '1234'  
where id_card = 1;
```

```

Error starting at line : 22 in command -
update bank_admin.card
set id_tip_card = 2
where id_card = 1
Error at Command Line : 22 Column : 19
Error report -
SQL Error: ORA-01031: insufficient privileges
01031. 00000 - "insufficient privileges"
*Cause:      An attempt was made to perform a database operation without
              the necessary privileges.
*Action:     Ask your database administrator or designated security
              administrator to grant you the necessary privileges

1 row updated.

```

- Personalul din departamentele de marketing și detectarea fraudelor are acces la toate tabelele din schemă, pentru a întocmi diverse rapoarte de interes. În plus, utilizatorul BANK_FRAUDA va putea actualiza statusul unui cont suspect de fraudă în „BLOCAT” (se va crea o procedură în schema adminului care va fi apelată de utilizatorul BANK_FRAUDA)

```

create role access_access_other_departments;
GRANT SELECT ON bank_admin.tip_cont to access_other_departments;
GRANT SELECT ON bank_admin.tip_card to access_other_departments;
GRANT SELECT ON bank_admin.status to access_other_departments;
GRANT SELECT ON bank_admin.banca to access_other_departments;
GRANT SELECT ON bank_admin.oras to access_other_departments;
GRANT SELECT ON bank_admin.tara to access_other_departments;
GRANT SELECT ON bank_admin.card to access_other_departments;
GRANT SELECT ON bank_admin.cont to access_other_departments;
GRANT SELECT ON bank_admin.client_cont to access_other_departments;
GRANT SELECT ON bank_admin.client to access_other_departments;

grant access_other_departments to bank_marketing, bank_frauda;

create role change_status;
grant update(id_tip_status) on bank_admin.cont to change_status;

grant change_status to bank_frauda;

```

	ROLE
1	ACCESS_OTHER_DEPARTMENTS
2	CHANGE_STATUS

```

create or replace procedure actualizare_status_cont(p_id_cont number)
as
begin
    update bank_admin.cont
    set id_tip_status = (select s.id_tip_status
                        from bank_admin.status s
                        where s.denumire like 'BLOCAT')
    where id_cont = p_id_cont;

end;
/




grant execute on actualizare_status_cont to bank_frauda;

```

```

--logat ca utilizatorul bank_rauda
execute bank_admin.actualizare_status_cont(1);

```




SQL

All Rows Fetched: 1 in 0.002 seconds

ID_CONT	ID_TIP_CONT	ID_TIP_STATUS	IBAN	DATA_CREATE	SOLD
1	1	1	3 SN26732036098651	12-SEP-18	299

6. Aplicațiile pe baza de date și securitatea datelor

a. Contextul aplicației

Va fi creat un context de aplicație care stabilește, ca măsură de securitate, introducerea sau modificarea conturilor de către consultanții bancari numai în intervalul orar de funcționare a băncii, între orele 8:00-17:00.

I. Crearea contextului

```
create context banca_ctx using proced_banca_ctx;
```

II. Crearea procedurii asociate contextului

```
create or replace procedure proced_banca_ctx
is
    v_ora number(3);
begin
    select to_number(to_char(sysdate, 'hh24')) into v_ora from dual;
    dbms_output.put_line('Este ora: ' || v_ora);
    if v_ora < 8 or v_ora > 17 then
        dbms_output.put_line('Sunteti in afara orelor de program. ');
        dbms_session.set_context('banca_ctx', 'ora_potrivita', 'nu');
    else
        dbms_session.set_context('banca_ctx', 'ora_potrivita', 'da');
    end if;
end;
/
```

III. Crearea unui trigger de logon care determină ca, la conectarea la baza de date, să fie stabilit automat contextul aplicației pentru sesiunea utilizatorului

```

create or replace trigger tr_after_logon
after logon on database
declare
    v_user varchar2(30);
begin
    v_user := sys_context('userenv', 'session_user');
    dbms_output.put_line('userul este ' || v_user );
    if lower(v_user) like '%bank_consultant1%' then
        proced_banca_ctx;
    end if;
end;
/

```

- IV. În schema administratorului bazei de date se va crea un trigger BEFORE INSERT OR UPDATE asupra tabelului CONT, care va prelua valoarea returnată de sys_context('banca_ctx', 'ora_potrivita') conform căreia va fi sau nu permisă operația LMD.

```

create or replace trigger tr_insert_update_cont
before insert or update on bank_admin.cont
for each row
declare
    v_poate varchar2(4);
begin
    v_poate := sys_context('banca_ctx', 'ora_potrivita');
    if (v_poate = 'nu') then
        dbms_output.put_line ('Nu aveti voie sa modificati conturile in afara orelor de program');
        :new.sold := :old.sold;
    end if;
end;
/

```

Tot în schema adminului am creat o procedură prin care se actualizează valoarea soldului unui cont.

```
create or replace procedure update_cont(p_id_cont number, p_id_sold number)
is
begin
    update bank_admin.cont
    set sold = p_id_sold
    where id_cont = p_id_cont;

exception
    when others then
        dbms_output.put_line('Eroare: ' || sqlerrm);
end;
/

grant execute on update_cont to bank_consultant1;
```

Din schema consultantului, am apelat procedura update_cont(1, 1000) la o oră în afara programului de lucru al băncii (ora 19:00).

```
Nu aveti voie sa modificati conturile in afara orelor de program
```

```
PL/SQL procedure successfully completed.
```

b. SQL Injection

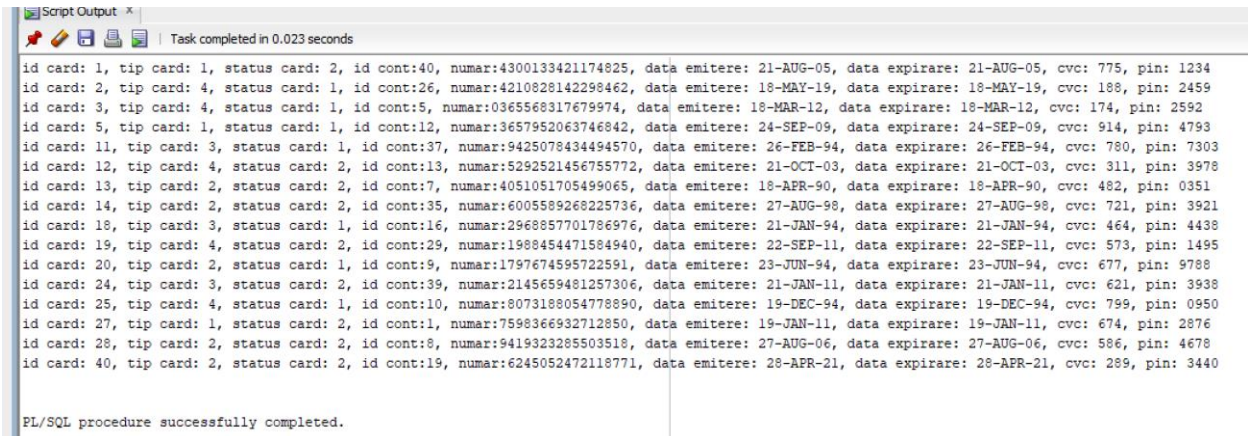
Am creat o procedură în schema utilizatorului bank_admin prin care se afisează toate cardurile emise după o anumită dată exactă furnizată ca parametru, utilizând SQL dinamic (EXECUTE IMMEDIATE). Această metodă prezintă vulnerabilități în fața atacurilor malițioase.

```
create or replace procedure afiseaza_carduri(p_data varchar2)
as
    TYPE vector_carduri IS TABLE OF bank_admin.card%ROWTYPE;
    v_vector vector_carduri;
begin
    EXECUTE IMMEDIATE 'select c.id_card, c.id_tip_card, c.id_tip_status, c.id_cont, c.numar,
c.data_emitere, c.data_expirare, c.cvc, c.pin
                        from bank_admin.card c
                        join bank_admin.cont co on(c.id_cont = co.id_cont)
                        join bank_admin.client_cont cc on(co.id_cont = cc.id_cont)
                        join bank_admin.client cl on(cl.id_client = cc.id_client)
                        where to_char(c.data_emitere,"DD-MON-YY") > ' || p_data || '
                        order by id_card'

    bulk collect into v_vector;
    for i in 1..v_vector.COUNT loop
        DBMS_OUTPUT.PUT_LINE('id card: ' || v_vector(i).id_card || ', tip card: ' ||
v_vector(i).id_tip_card
        || ', status card: ' || v_vector(i).id_tip_status || ', id cont:' || v_vector(i).id_cont ||
        ', numar:' || v_vector(i).numar || ', data emitere: ' || v_vector(i).data_emitere || ', data
expirare: ' || v_vector(i).data_expirare
        || ', cvc: ' || v_vector(i).cvc || ', pin: ' || v_vector(i).pin);
    end loop
end;
/
```


Am executat procedura ca bank_consultant1, după ce acesta a primit privilegiu asupra procedurii. Prima dată este furnizat un parametru corect pentru dată, fiind afișate cardurile emise după data de 17 mai 2003.

```
execute bank_admin.afiseaza_carduri('17-MAY-03');
```



The screenshot shows a 'Script Output' window with a toolbar at the top. Below the toolbar, it says 'Task completed in 0.023 seconds'. The main area contains a list of 20 cards, each with fields: id card, tip card, status card, id cont, numar, data emiter, data expirare, cvc, and pin. The cards are numbered 1 through 20. At the bottom, it says 'PL/SQL procedure successfully completed.'

```
id card: 1, tip card: 1, status card: 2, id cont:40, numar:4300133421174825, data emiter: 21-AUG-05, data expirare: 21-AUG-05, cvc: 775, pin: 1234
id card: 2, tip card: 4, status card: 1, id cont:26, numar:4210828142298462, data emiter: 18-MAY-19, data expirare: 18-MAY-19, cvc: 188, pin: 2459
id card: 3, tip card: 4, status card: 1, id cont:5, numar:0365568317679974, data emiter: 18-MAR-12, data expirare: 18-MAR-12, cvc: 174, pin: 2592
id card: 5, tip card: 1, status card: 1, id cont:12, numar:3657952063746842, data emiter: 24-SEP-09, data expirare: 24-SEP-09, cvc: 914, pin: 4793
id card: 11, tip card: 3, status card: 1, id cont:37, numar:94250784344494570, data emiter: 26-FEB-94, data expirare: 26-FEB-94, cvc: 780, pin: 7303
id card: 12, tip card: 4, status card: 2, id cont:13, numar:5292521456755772, data emiter: 21-OCT-03, data expirare: 21-OCT-03, cvc: 311, pin: 3978
id card: 13, tip card: 2, status card: 2, id cont:7, numar:4051051705499065, data emiter: 18-APR-90, data expirare: 18-APR-90, cvc: 482, pin: 0351
id card: 14, tip card: 2, status card: 2, id cont:35, numar:6005589268225736, data emiter: 27-AUG-98, data expirare: 27-AUG-98, cvc: 721, pin: 3921
id card: 18, tip card: 3, status card: 1, id cont:16, numar:2968857701786976, data emiter: 21-JAN-94, data expirare: 21-JAN-94, cvc: 464, pin: 4438
id card: 19, tip card: 4, status card: 2, id cont:29, numar:1988454471584940, data emiter: 22-SEP-11, data expirare: 22-SEP-11, cvc: 573, pin: 1495
id card: 20, tip card: 2, status card: 1, id cont:9, numar:1797674595722591, data emiter: 23-JUN-94, data expirare: 23-JUN-94, cvc: 677, pin: 9788
id card: 24, tip card: 3, status card: 2, id cont:39, numar:2145659481257306, data emiter: 21-JAN-11, data expirare: 21-JAN-11, cvc: 621, pin: 3938
id card: 25, tip card: 4, status card: 1, id cont:10, numar:8073188054778890, data emiter: 19-DEC-94, data expirare: 19-DEC-94, cvc: 799, pin: 0950
id card: 27, tip card: 1, status card: 2, id cont:1, numar:7598366932712850, data emiter: 19-JAN-11, data expirare: 19-JAN-11, cvc: 674, pin: 2876
id card: 28, tip card: 2, status card: 2, id cont:8, numar:9419323285503518, data emiter: 27-AUG-06, data expirare: 27-AUG-06, cvc: 586, pin: 4678
id card: 40, tip card: 2, status card: 2, id cont:19, numar:6245052472118771, data emiter: 28-APR-21, data expirare: 28-APR-21, cvc: 289, pin: 3440

PL/SQL procedure successfully completed.
```

Un exemplu de atac malițios poate fi următorul, prin care se obțin informații despre toate cardurile din baza de date.

```
execute bank_admin.afiseaza_carduri('17-MAY-03" OR 1=1--');
```

```
Script Output x
Task completed in 0.021 seconds

id card: 1, tip card: 1, status card: 2, id cont:40, numar:4300133421174825, data emitere: 21-AUG-05, data expirare: 21-AUG-05, cvc: 775, pin: 1234
id card: 2, tip card: 4, status card: 1, id cont:26, numar:4210828142298462, data emitere: 18-MAY-19, data expirare: 18-MAY-19, cvc: 188, pin: 2459
id card: 3, tip card: 4, status card: 1, id cont:5, numar:0365568317679974, data emitere: 18-MAR-12, data expirare: 18-MAR-12, cvc: 174, pin: 2592
id card: 4, tip card: 1, status card: 2, id cont:40, numar:2458651400800579, data emitere: 02-APR-98, data expirare: 02-APR-98, cvc: 801, pin: 0920
id card: 5, tip card: 1, status card: 1, id cont:12, numar:3657952063746842, data emitere: 24-SEP-09, data expirare: 24-SEP-09, cvc: 914, pin: 4793
id card: 6, tip card: 4, status card: 2, id cont:4, numar:3015443114872683, data emitere: 15-JAN-21, data expirare: 15-JAN-21, cvc: 858, pin: 2640
id card: 7, tip card: 4, status card: 2, id cont:16, numar:8457433511838331, data emitere: 06-DEC-09, data expirare: 06-DEC-09, cvc: 812, pin: 8208
id card: 8, tip card: 3, status card: 1, id cont:11, numar:4931132637651661, data emitere: 11-FEB-93, data expirare: 11-FEB-93, cvc: 270, pin: 4447
id card: 9, tip card: 4, status card: 2, id cont:10, numar:7662278547178375, data emitere: 09-APR-16, data expirare: 09-APR-16, cvc: 509, pin: 8812
id card: 10, tip card: 3, status card: 1, id cont:11, numar:2707339084182844, data emitere: 04-MAR-06, data expirare: 04-MAR-06, cvc: 441, pin: 4592
id card: 11, tip card: 3, status card: 1, id cont:37, numar:9425078434494570, data emitere: 26-FEB-94, data expirare: 26-FEB-94, cvc: 780, pin: 7303
id card: 12, tip card: 4, status card: 2, id cont:13, numar:5292521456755772, data emitere: 21-OCT-03, data expirare: 21-OCT-03, cvc: 311, pin: 3978
id card: 13, tip card: 2, status card: 2, id cont:7, numar:4051051705499065, data emitere: 18-APR-90, data expirare: 18-APR-90, cvc: 482, pin: 0351
id card: 14, tip card: 2, status card: 2, id cont:35, numar:6005589268225736, data emitere: 27-AUG-98, data expirare: 27-AUG-98, cvc: 721, pin: 3921
id card: 15, tip card: 2, status card: 1, id cont:37, numar:4175637603846875, data emitere: 17-MAY-03, data expirare: 17-MAY-03, cvc: 282, pin: 3839
id card: 16, tip card: 4, status card: 2, id cont:9, numar:0398642661393479, data emitere: 05-FEB-00, data expirare: 05-FEB-00, cvc: 736, pin: 5467
id card: 17, tip card: 3, status card: 2, id cont:27, numar:3739564367884864, data emitere: 11-APR-09, data expirare: 11-APR-09, cvc: 330, pin: 5625
id card: 18, tip card: 3, status card: 1, id cont:16, numar:2968857701786976, data emitere: 21-JAN-94, data expirare: 21-JAN-94, cvc: 464, pin: 4438
id card: 19, tip card: 4, status card: 2, id cont:29, numar:1988454471584940, data emitere: 22-SEP-11, data expirare: 22-SEP-11, cvc: 573, pin: 1495
id card: 20, tip card: 2, status card: 1, id cont:9, numar:1797674595722591, data emitere: 23-JUN-94, data expirare: 23-JUN-94, cvc: 677, pin: 9788
id card: 21, tip card: 4, status card: 1, id cont:9, numar:3119836209117029, data emitere: 14-DEC-00, data expirare: 14-DEC-00, cvc: 710, pin: 4641
id card: 22, tip card: 3, status card: 2, id cont:12, numar:9873751807322337, data emitere: 06-MAR-01, data expirare: 06-MAR-01, cvc: 226, pin: 9650
id card: 23, tip card: 3, status card: 2, id cont:29, numar:7974277682159252, data emitere: 11-MAR-92, data expirare: 11-MAR-92, cvc: 997, pin: 1206
id card: 24, tip card: 3, status card: 2, id cont:39, numar:2145659481257306, data emitere: 21-JAN-11, data expirare: 21-JAN-11, cvc: 621, pin: 3938
id card: 25, tip card: 4, status card: 1, id cont:10, numar:8073188054778890, data emitere: 19-DEC-94, data expirare: 19-DEC-94, cvc: 799, pin: 0950
id card: 26, tip card: 2, status card: 2, id cont:12, numar:6006442464553363, data emitere: 07-JAN-95, data expirare: 07-JAN-95, cvc: 339, pin: 8024
id card: 27, tip card: 1, status card: 2, id cont:1, numar:7598366932712850, data emitere: 19-JAN-11, data expirare: 19-JAN-11, cvc: 674, pin: 2876
id card: 28, tip card: 2, status card: 2, id cont:8, numar:9419323285503518, data emitere: 27-AUG-06, data expirare: 27-AUG-06, cvc: 586, pin: 4678
id card: 29, tip card: 1, status card: 1, id cont:19, numar:3894849228203228, data emitere: 05-JUN-11, data expirare: 05-JUN-11, cvc: 914, pin: 0678
id card: 30, tip card: 1, status card: 2, id cont:17, numar:0076900826263994, data emitere: 09-AUG-95, data expirare: 09-AUG-95, cvc: 943, pin: 9046
```

Un alt exemplu de SQL Injection poate afișa numai cardurile care îndeplinesc o anumită condiție, în cazul de față cu soldul mai mare decat 5000, informație care nu ar trebui furnizată.

```
execute bank_admin.afiseaza_carduri('17-MAY-03" and c.id_cont in (select id_cont from bank_admin.cont where sold > 5000)--');
```

```
id card: 1, tip card: 1, status card: 2, id cont:40, numar:4300133421174825, data emitere: 21-AUG-05, data expirare: 21-AUG-05, cvc: 775, pin: 1234
id card: 5, tip card: 1, status card: 1, id cont:12, numar:3657952063746842, data emitere: 24-SEP-09, data expirare: 24-SEP-09, cvc: 914, pin: 4793
id card: 13, tip card: 2, status card: 2, id cont:7, numar:4051051705499065, data emitere: 18-APR-90, data expirare: 18-APR-90, cvc: 482, pin: 0351
id card: 25, tip card: 4, status card: 1, id cont:10, numar:8073188054778890, data emitere: 19-DEC-94, data expirare: 19-DEC-94, cvc: 799, pin: 0950
id card: 28, tip card: 2, status card: 2, id cont:8, numar:9419323285503518, data emitere: 27-AUG-06, data expirare: 27-AUG-06, cvc: 586, pin: 4678
id card: 40, tip card: 2, status card: 2, id cont:19, numar:6245052472118771, data emitere: 28-APR-21, data expirare: 28-APR-21, cvc: 289, pin: 3440

PL/SQL procedure successfully completed.
```

Varianta sigură prin care sunt prevenite atacurile de tip SQL Injection este prin utilizarea variabilelor de legătură (bind variables).

```
create or replace procedure afiseaza_carduri_safe(p_data varchar2)
as
    TYPE vector_carduri IS TABLE OF bank_admin.card%ROWTYPE;
    v_vector vector_carduri;
```

```

begin
    EXECUTE IMMEDIATE 'select c.id_card, c.id_tip_card, c.id_tip_status, c.id_cont, c.numar,
c.data_emitere, c.data_expirare, c.cvc, c.pin
        from bank_admin.card c
        join bank_admin.cont co on(c.id_cont = co.id_cont)
        join bank_admin.client_cont cc on(co.id_cont = cc.id_cont)
        join bank_admin.client cl on(cl.id_client = cc.id_client)
        where to_char(c.data_emitere,"DD-MON-YY") > :data
        order by id_card'

    bulk collect into v_vector
    using p_data;
    for i in 1..v_vector.COUNT loop
        DBMS_OUTPUT.PUT_LINE('id card: ' || v_vector(i).id_card || ', tip card: ' ||
v_vector(i).id_tip_card
        || ', status card: ' || v_vector(i).id_tip_status || ', id cont:' || v_vector(i).id_cont ||
        ', numar:' || v_vector(i).numar || ', data emitere: ' || v_vector(i).data_emitere || ', data
expirare: ' || v_vector(i).data_expirare
        || ', cvc: ' || v_vector(i).cvc || ', pin: ' || v_vector(i).pin);
    end loop;
end;

```

Prin această metodă sunt afișate numai datele de interes ce respectă condiția din interogarea dinamică din procedură.

```

execute bank_admin.afiseaza_carduri_safe('17-MAY-03')
execute bank_admin.afiseaza_carduri_safe('17-MAY-03" OR 1=1--');
execute bank_admin.afiseaza_carduri_safe('17-MAY-03" and c.id_cont in (select id_cont from
bank_admin.cont where sold > 5000)--');

```

```
Script Output x
Task completed in 0.025 seconds

id card: 1, tip card: 1, status card: 2, id cont:40, numar:4300133421174825, data emitere: 21-AUG-05, data expirare: 21-AUG-05, cvc: 775, pin: 1234
id card: 2, tip card: 4, status card: 1, id cont:26, numar:4210828142298462, data emitere: 18-MAY-19, data expirare: 18-MAY-19, cvc: 188, pin: 2459
id card: 3, tip card: 4, status card: 1, id cont:5, numar:0365568317679974, data emitere: 18-MAR-12, data expirare: 18-MAR-12, cvc: 174, pin: 2592
id card: 5, tip card: 1, status card: 1, id cont:12, numar:3657952063746842, data emitere: 24-SEP-09, data expirare: 24-SEP-09, cvc: 914, pin: 4793
id card: 11, tip card: 3, status card: 1, id cont:37, numar:9425078434494570, data emitere: 26-FEB-94, data expirare: 26-FEB-94, cvc: 780, pin: 7303
id card: 12, tip card: 4, status card: 2, id cont:13, numar:5292521456755772, data emitere: 21-OCT-03, data expirare: 21-OCT-03, cvc: 311, pin: 3978
id card: 13, tip card: 2, status card: 2, id cont:7, numar:4051051705499065, data emitere: 18-APR-90, data expirare: 18-APR-90, cvc: 482, pin: 0351
id card: 14, tip card: 2, status card: 2, id cont:35, numar:6005589268225736, data emitere: 27-AUG-98, data expirare: 27-AUG-98, cvc: 721, pin: 3921
id card: 18, tip card: 3, status card: 1, id cont:16, numar:2968857701786976, data emitere: 21-JAN-94, data expirare: 21-JAN-94, cvc: 464, pin: 4438
id card: 19, tip card: 4, status card: 2, id cont:29, numar:1988454471584940, data emitere: 22-SEP-11, data expirare: 22-SEP-11, cvc: 573, pin: 1495
id card: 20, tip card: 2, status card: 1, id cont:9, numar:1797674595722591, data emitere: 23-JUN-94, data expirare: 23-JUN-94, cvc: 677, pin: 9788
id card: 24, tip card: 3, status card: 2, id cont:39, numar:2145659481257306, data emitere: 21-JAN-11, data expirare: 21-JAN-11, cvc: 621, pin: 3938
id card: 25, tip card: 4, status card: 1, id cont:10, numar:8073188054778890, data emitere: 19-DEC-94, data expirare: 19-DEC-94, cvc: 799, pin: 0950
id card: 27, tip card: 1, status card: 2, id cont:1, numar:7598366932712850, data emitere: 19-JAN-11, data expirare: 19-JAN-11, cvc: 674, pin: 2876
id card: 28, tip card: 2, status card: 2, id cont:8, numar:9419323285503518, data emitere: 27-AUG-06, data expirare: 27-AUG-06, cvc: 586, pin: 4678
id card: 40, tip card: 2, status card: 2, id cont:19, numar:6245052472118771, data emitere: 28-APR-21, data expirare: 28-APR-21, cvc: 289, pin: 3440
```