

Aplicatie de evaluare

Securitatea Bazelor de Date

CareConnect

Autor: Murariu Andrei

Rezumat

Lucrarea prezintă un model de bază de date medicală securizată, CareConnect, proiectat în Oracle Database și organizat în jurul entităților pacient, personal medical, fișă medicală și departament. Securitatea datelor sensibile (CNP) este asigurată prin criptare cu pachetul `DBMS_CRYPTO` folosind algoritmul AES-256 în modul CBC și prin gestionarea cheilor într-un tabel dedicat, cu mecanisme de rotire și auditare. Controlul accesului este implementat prin Role-Based Access Control (RBAC), Virtual Private Database (VPD) cu securitate la nivel de rând și view-uri cu mascare de date, completate de auditare standard, trigger-i de auditare și Fine-Grained Auditing (FGA). Toate aceste mecanisme sunt integrate și demonstrate practic într-o aplicație CLI Python.

Cuprins

1	CareConnect	3
1.1	Prezentarea modelului proiectat	3
1.1.1	Reguli de business	3
1.2	Diagrama conceptuală	4
1.3	Schemele relaționale	5
1.4	Crearea tabelor	5
1.5	Reguli de securitate	6
1.5.1	Controlul accesului	7
1.5.2	Criptarea datelor	9
1.5.3	Auditarea	11
1.5.4	Prevenirea atacurilor	13
1.5.5	Mascarea datelor	14

Capitolul 1

CareConnect

1.1 Prezentarea modelului proiectat

Arhiva repository: https://github.com/iamxorum/Master_Unibuc

Baza de date CareConnect este un sistem medical care gestionează informații despre pacienți, personalul medical și fișele medicale asociate. Modelul este proiectat pentru a demonstra mecanisme avansate de securitate în Oracle Database, respectând principiile de confidențialitate și integritate a datelor medicale sensibile.

Sistemul este structurat pe patru entități principale:

- **DEPARTAMENT** - reprezintă departamentele medicale ale instituției (ex: Cardiologie, Neurologie)
- **PERSONAL_MEDICAL** - angajații sistemului medical (medici, asistenți, personal de recepție, administratori)
- **PACIENT** - pacienții înregistrați în sistem
- **FISA_MEDICALA** - fișele medicale asociate pacienților, create de medici

1.1.1 Reguli de business

Modelul respectă următoarele reguli de business:

1. Fiecare pacient are un CNP unic, care este criptat în baza de date folosind AES-256
2. Fiecare fișă medicală este asociată unui pacient și unui medic responsabil
3. Fișele medicale au un nivel de confidențialitate (1, 2 sau 3) care determină accesul utilizatorilor
4. Personalul medical este organizat pe departamente și are un rol (MEDIC, ASISTENT, RECEPȚIE, ADMIN) cu un grad de acces corespunzător (1-4)

5. Fiecare membru al personalului medical are un cont Oracle Database asociat pentru autentificare

1.2 Diagrama conceptuală

Diagrama conceptuală a modelului este prezentată în Figura.

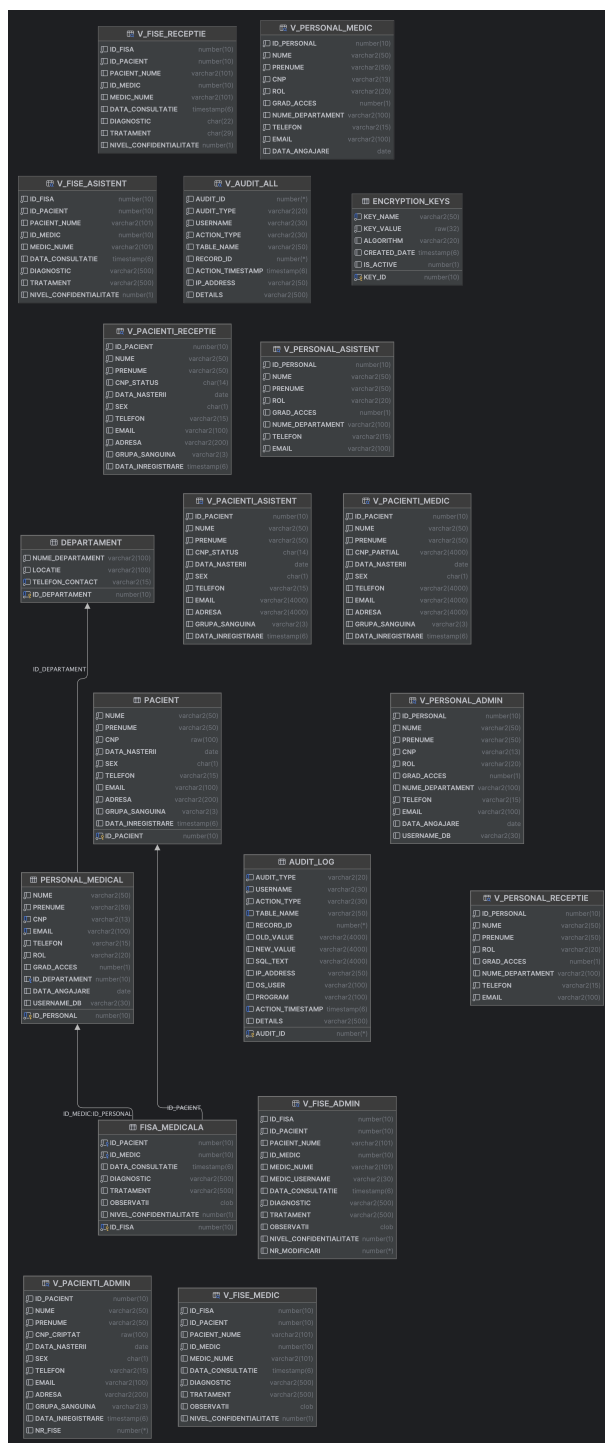


Figura 1.1: Diagrama CareConnect

Aceasta ilustrează entitățile principale și relațiile dintre ele:

- **DEPARTAMENT** ↔ **PERSONAL_MEDICAL**: relație 1:N (un departament are mai mulți angajați)
- **PERSONAL_MEDICAL** ↔ **FISA_MEDICALA**: relație 1:N (un medic poate crea mai multe fișe)
- **PACIENT** ↔ **FISA_MEDICALA**: relație 1:N (un pacient poate avea mai multe fișe medicale)

1.3 Schemele relaționale

Schema relațională a bazei de date este următoarea:

DEPARTAMENT

PERSONAL_MEDICAL • FK: id_departament → DEPARTAMENT(id_departament)

- Constraint: rol ∈ {'MEDIC', 'ASISTENT', 'ADMIN', 'RECEPTIE'}
- Constraint: grad_acces ∈ {0, 1, 2, 3, 4}

PACIENT • Constraint: sex ∈ {'M', 'F'}

- Constraint: grupa_sanguina ∈ {'A+', 'A-', 'B+', 'B-', 'AB+', 'AB-', 'O+', 'O-'}
- Notă: cnp este de tip RAW(100) - stocat criptat

FISA_MEDICALA • FK: id_pacient → PACIENT(id_pacient)

- FK: id_medic → PERSONAL_MEDICAL(id_personal)
- Constraint: nivel_confidentialitate ∈ {1, 2, 3}

ENCRYPTION_KEYS • Tabel auxiliar pentru stocarea cheilor de criptare AES-

256

1.4 Crearea tabelelor

Scriptul de creare a tabelelor este disponibil în fișierul.

```

CREATE TABLE careconnect.departament (
    id_departament NUMBER(10) PRIMARY KEY,
    nume_departament VARCHAR2(100) NOT NULL,
    locatie VARCHAR2(100) NOT NULL,
    telefon_contact VARCHAR2(15) NOT NULL UNIQUE
);

CREATE TABLE careconnect.personal_medical (
    id_personal NUMBER(10) PRIMARY KEY,
    nume VARCHAR2(50) NOT NULL,
    prenume VARCHAR2(50) NOT NULL,
    cnp VARCHAR2(13) NOT NULL UNIQUE,
    email VARCHAR2(100) NOT NULL UNIQUE,
    telefon VARCHAR2(15) NOT NULL,
    rol VARCHAR2(20) NOT NULL CHECK (rol IN ('MEDIC', 'ASISTENT', 'ADMIN', 'RECEPTIE')),
    grad_acces NUMBER(1) DEFAULT 1 CHECK (grad_acces IN (0, 1, 2, 3, 4)),
    id_departament NUMBER(10) REFERENCES careconnect.departament(id_departament),
    data_angajare DATE DEFAULT SYSDATE,
    username_db VARCHAR2(30)
);

CREATE TABLE careconnect.pacient (
    id_pacient NUMBER(10) PRIMARY KEY,
    nume VARCHAR2(50) NOT NULL,
    prenume VARCHAR2(50) NOT NULL,
    cnp RAW(100) NOT NULL,
    data_nasterii DATE NOT NULL,
    sex CHAR(1) NOT NULL CHECK (sex IN ('M', 'F')),
    telefon VARCHAR2(15) NOT NULL,
    email VARCHAR2(100),
    adresa VARCHAR2(200) NOT NULL,
    grupa_sanguina VARCHAR2(3) CHECK (grupa_sanguina IN ('A+', 'A-', 'B+', 'B-', 'AB+', 'AB-', 'O+', 'O-')),
    data_inregistrare TIMESTAMP DEFAULT SYSTIMESTAMP
);

CREATE TABLE careconnect.fisa_medicala (
    id_fisa NUMBER(10) PRIMARY KEY,
    id_pacient NUMBER(10) NOT NULL REFERENCES careconnect.pacient(id_pacient),
    id_medec NUMBER(10) NOT NULL REFERENCES careconnect.personal_medical(id_personal),
    data_consultatie TIMESTAMP DEFAULT SYSTIMESTAMP,
    diagnostic VARCHAR2(500) NOT NULL,
    tratament VARCHAR2(500),
    observatii CLOB,
    nivel_confidentialitate NUMBER(1) DEFAULT 1 CHECK (nivel_confidentialitate IN (1, 2, 3))
);

CREATE SEQUENCE careconnect.seq_departament START WITH 1 INCREMENT BY 1;
CREATE SEQUENCE careconnect.seq_personal START WITH 1 INCREMENT BY 1;
CREATE SEQUENCE careconnect.seq_pacient START WITH 1 INCREMENT BY 1;
CREATE SEQUENCE careconnect.seq_fisa START WITH 1 INCREMENT BY 1;

CREATE TABLE careconnect.encryption_keys (
    key_id NUMBER(10) PRIMARY KEY,
    key_name VARCHAR2(50) NOT NULL UNIQUE,
    key_value RAW(32) NOT NULL,
    algorithm VARCHAR2(20) DEFAULT 'AES256',
    created_date TIMESTAMP DEFAULT SYSTIMESTAMP,
    is_active NUMBER(1) DEFAULT 1
);

```

Figura 1.2: Schema CareConnect

Structura cheilor primare și a relațiilor este următoarea:

- **Chei primare:** id_departament, id_personal, id_pacient, id_fisa, key_id
- **Secvențe:** seq_departament, seq_personal, seq_pacient, seq_fisa
- **Relații:**
 - PERSONAL_MEDICAL.id_departament → DEPARTAMENT.id_departament
 - FISA_MEDICALA.id_pacient → PACIENT.id_pacient
 - FISA_MEDICALA.id_medec → PERSONAL_MEDICAL.id_personal

1.5 Reguli de securitate

Fiecare personal_medical are propriul user in Database creat cu privilegiile acordate ulterior pe baza gradului/rolului de acces dat la creare; pentru demo avem 4 înregistrări, 1 pentru fiecare rol și mai jos se poate observa existența lor ca useri in baza de date; adăugarea unor noi membrii în personalul medical creează un alt user respectiv in baza de date (se poate observa și în audit log).

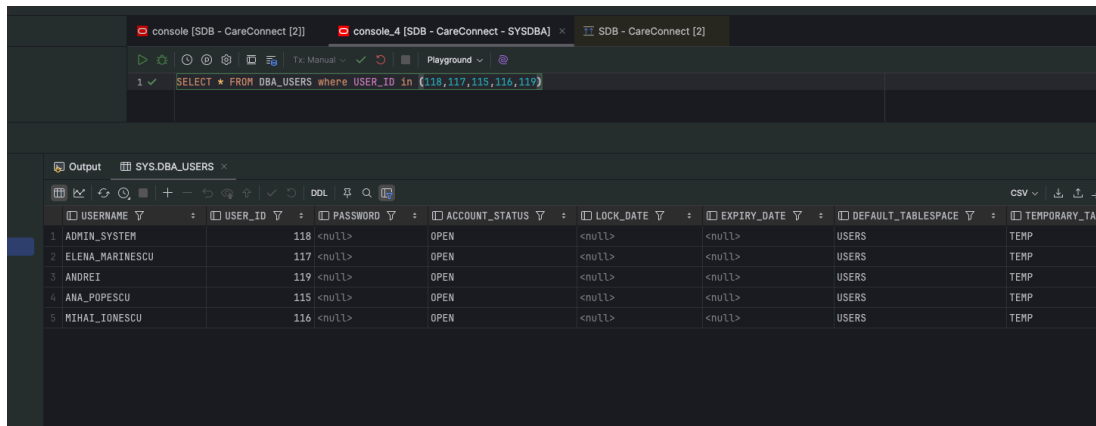


Figura 1.3: DB Users

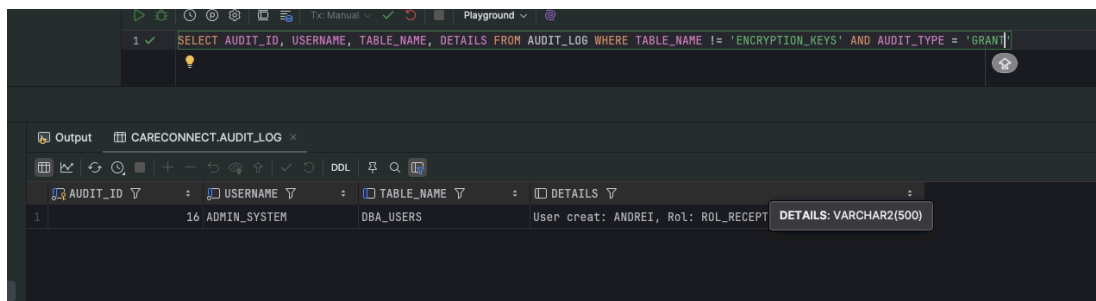


Figura 1.4: DB Users Audit

1.5.1 Controlul accesului

1. **Role-Based Access Control (RBAC)**: Sistemul definește patru roluri ierarhice:

- **ROL_RECEPTIE** (grad_acces=1): poate înregistra pacienți noi, citește date limitate
- **ROL_ASISTENT** (grad_acces=2): moștenește privilegiile recepției, poate citi fișe medicale cu nivel ≤ 2
- **ROL_MEDIC** (grad_acces=3): moștenește privilegiile asistentului, poate crea/-modifica fișe medicale, poate decripta CNP
- **ROL_ADMIN** (grad_acces=4): acces complet, poate gestiona personalul, roti chei de criptare, accesează audit logs

2. **Virtual Private Database (VPD)**: Implementat prin Row-Level Security (RLS) pentru filtrarea automată a fișelor medicale pe baza nivelului de confidențialitate și gradului de acces al utilizatorului

3. **Views cu mascare**: Fiecare rol are view-uri dedicate care maschează datele sensibile (CNP, email, telefon, adresă) conform nivelului de acces

Mai jos se pot observa exemple de acces diferite bazate pe fiecare rol/grad de acces:



```
CareConnect | ELENA_MARINESCU (RECEPȚIE) | Grad: 1
Ping: 2ms

Fișe Medicale



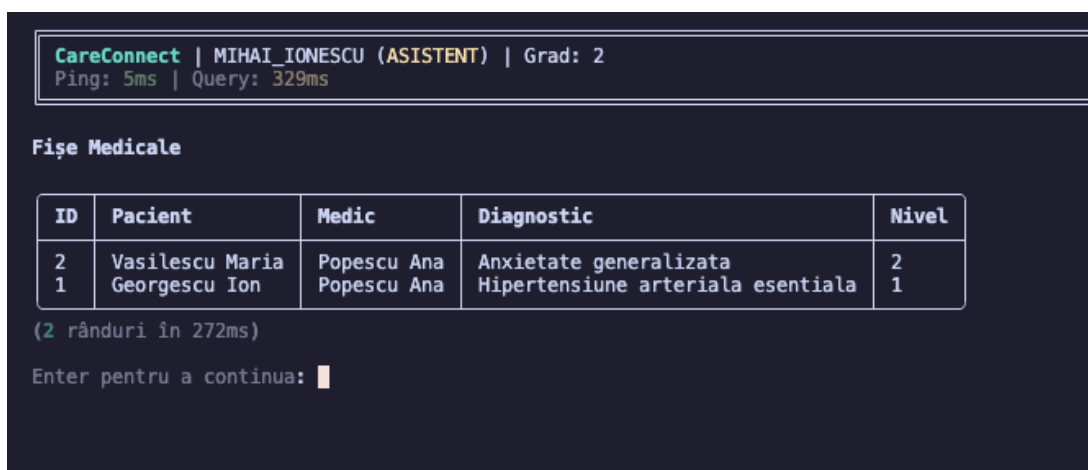
| ID | Pacient       | Medic       | Diagnostic           | Nivel |
|----|---------------|-------------|----------------------|-------|
| 1  | Georgescu Ion | Popescu Ana | Consultație medicală | 1     |



(1 rânduri în 329ms)

Enter pentru a continua: █
```

Figura 1.5: Fisa medical - acces grad 1



```
CareConnect | MIHAI_IONESCU (ASISTENT) | Grad: 2
Ping: 5ms | Query: 329ms

Fișe Medicale



| ID | Pacient         | Medic       | Diagnostic                        | Nivel |
|----|-----------------|-------------|-----------------------------------|-------|
| 2  | Vasilescu Maria | Popescu Ana | Anxietate generalizata            | 2     |
| 1  | Georgescu Ion   | Popescu Ana | Hipertensiune arteriala esentiala | 1     |



(2 rânduri în 272ms)

Enter pentru a continua: █
```

Figura 1.6: Fisa medical - acces grad 2



Figura 1.7: Fisa medical - acces grad 3

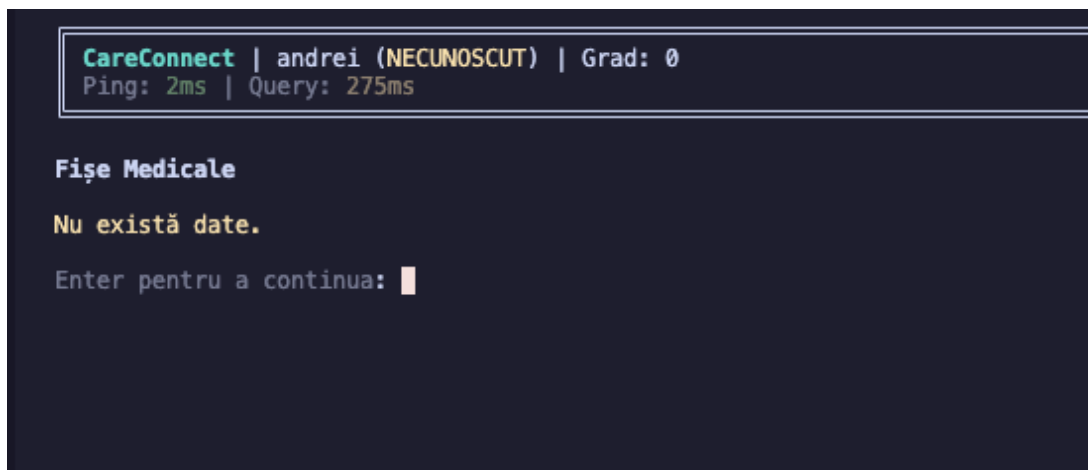


Figura 1.8: Fisa medical - acces grad 0

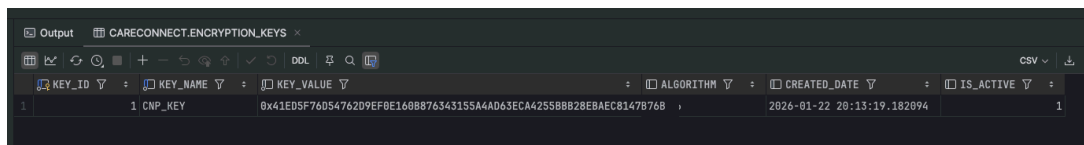
1.5.2 Criptarea datelor

1. **Criptare CNP:** CNP-ul pacienților este criptat folosind DBMS_CRYPTO cu algoritmul AES-256 în modul CBC

ID_PACIENT	NUME	PRENUME	CNP (UUID)	DATA_NASTERII	SEX	TELEFON
1	Georgescu	Ion	8D92EA9C35EB63DC2253A1B895C4964A	1985-04-15	M	073100000
2	Vasilescu	Maria	FAF07DBE38995DF884CAB10B846DF7AE	1990-05-20	F	073100000
3	Dumitrescu	Andrei	78E6811DEABEB017712C8BDCB84479FA	1995-06-30	M	073100000

Figura 1.9: CNP Criptat

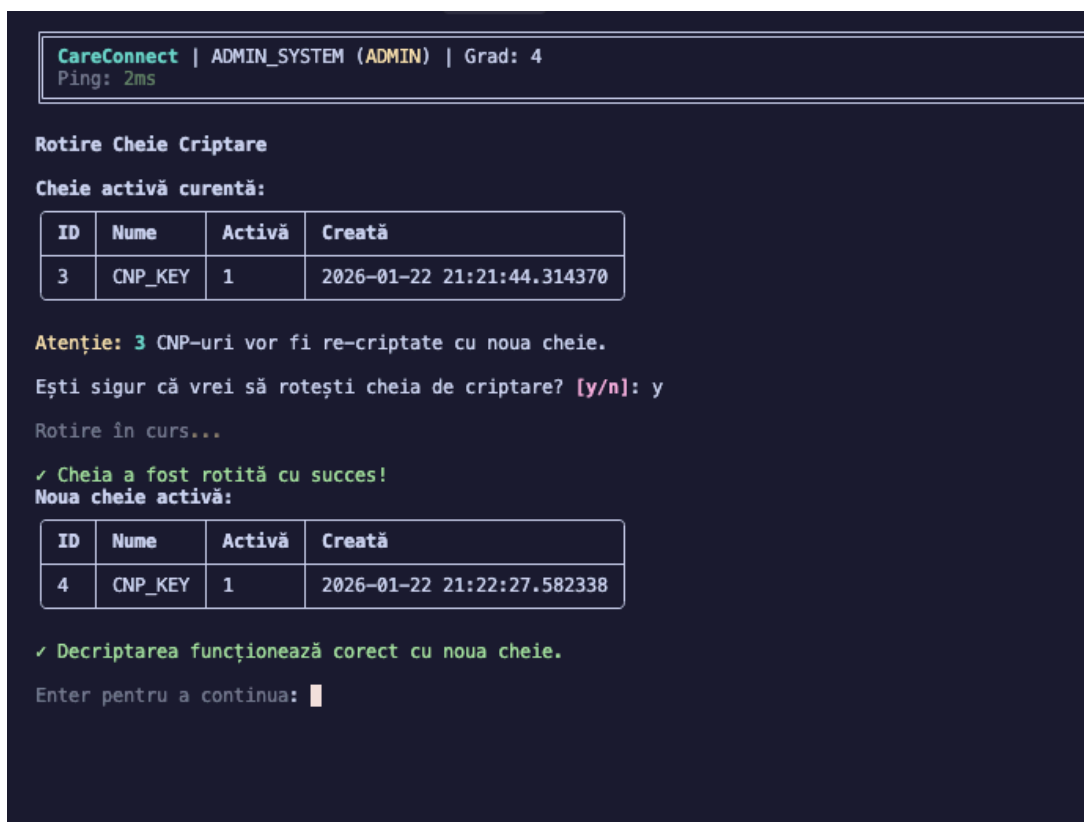
2. **Stocare chei:** Cheile de criptare sunt stocate într-un tabel separat (ENCRYPTION_KEYS) cu acces restricționat



KEY_ID	KEY_NAME	KEY_VALUE	ALGORITHM	CREATED_DATE	IS_ACTIVE
1	CNP_KEY	0x41ED5F76D54762D9EF8E168B876343155A4AD63ECA42558BB28E8AEC81478768	AES256	2026-01-22 20:13:19.182094	1

Figura 1.10: Chei de criptare AES256-CBC

3. **Rotire chei:** Procedură implementată pentru rotirea periodică a cheilor de criptare



```
CareConnect | ADMIN_SYSTEM (ADMIN) | Grad: 4
Ping: 2ms

Rotire Cheie Criptare

Cheie activă curentă:



| ID | Nume    | Activă | Creată                     |
|----|---------|--------|----------------------------|
| 3  | CNP_KEY | 1      | 2026-01-22 21:21:44.314370 |



Atenție: 3 CNP-uri vor fi re-criptate cu noua cheie.

Ești sigur că vrei să rotești cheia de criptare? [y/n]: y

Rotire în curs...

✓ Cheia a fost rotită cu succes!
Noua cheie activă:



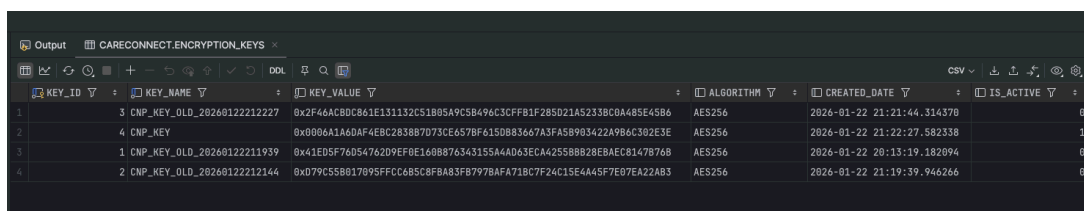
| ID | Nume    | Activă | Creată                     |
|----|---------|--------|----------------------------|
| 4  | CNP_KEY | 1      | 2026-01-22 21:22:27.582338 |



✓ Decriptarea funcționează corect cu noua cheie.

Enter pentru a continua:
```

Figura 1.11: Rotire chei de criptare



KEY_ID	KEY_NAME	KEY_VALUE	ALGORITHM	CREATED_DATE	IS_ACTIVE
3	CNP_KEY_OLD_20260122212227	0x2F4AAC8DC861E131132C51805A9C5B496C3CFFB1F285D21A52338CA485E45B6	AES256	2026-01-22 21:21:44.314370	0
4	CNP_KEY	0x0006A1A6DAF4EBC2838B7D73CE657BF6150B83667A3FA5B903422A9B6C302E3E	AES256	2026-01-22 21:22:27.582338	1
1	CNP_KEY_OLD_20260122211939	0x41ED5F76D54762D9EF8E168B876343155A4AD63ECA42558BB28E8AEC81478768	AES256	2026-01-22 20:13:19.182094	0
2	CNP_KEY_OLD_20260122212144	0xD79C558B17895FFCC6B5C8F8A83F877978AFA71BC7F24C15E4A45F7E07EA22AB3	AES256	2026-01-22 21:19:39.946266	0

Figura 1.12: Lista chei

AUDIT_ID	USERNAME	TABLE_NAME	DETAILS
33	ADMIN_SYSTEM	ENCRYPTION_KEYS	Cheie rotita cu succes
44	ADMIN_SYSTEM	ENCRYPTION_KEYS	Cheie rotita cu succes
25	CARECONNECT	ENCRYPTION_KEYS	Cheie rotita cu succes

Figura 1.13: Audit Rotire chei

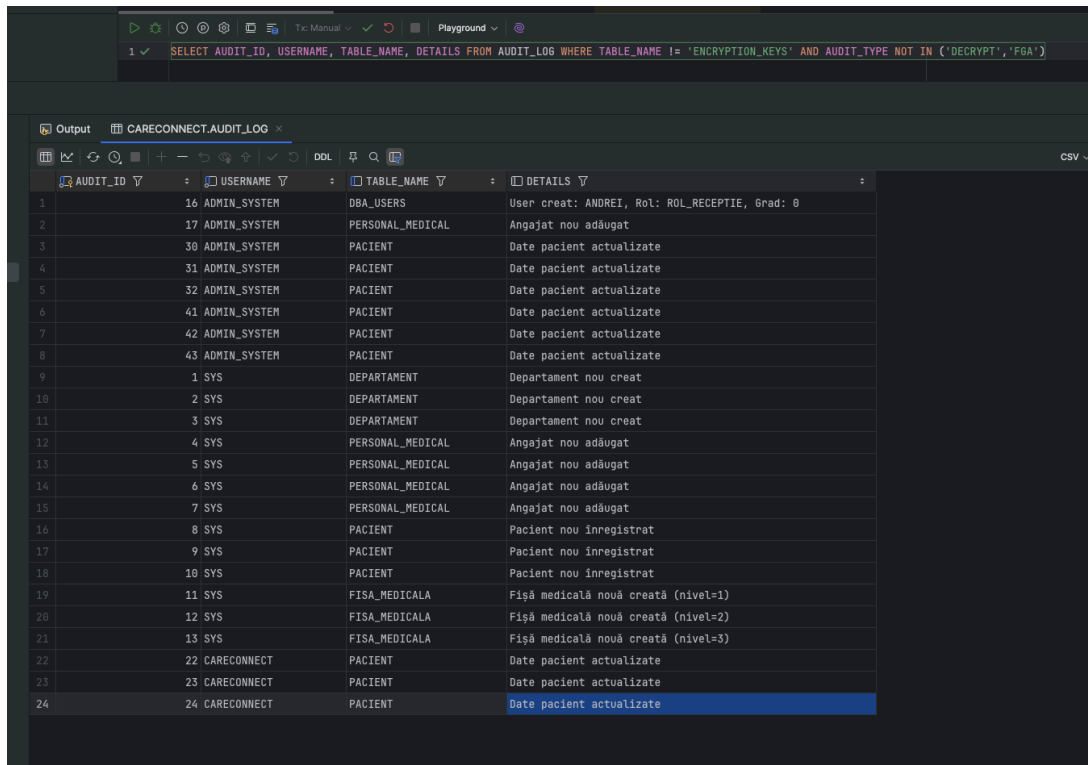
4. **Audit decriptări:** Toate decriptările CNP-ului sunt înregistrate în audit log

AUDIT_ID	USERNAME	TABLE_NAME	DETAILS
37	ADMIN_SYSTEM	PACIENT	CNP decriptat

Figura 1.14: Audit decriptare CNP

1.5.3 Auditarea

1. **Trigger-i de auditare:** Implementați pentru înregistrarea modificărilor (INSERT/UPDATE/DELETE) pe tabelele principale cu capturarea valorilor vechi și noi



The screenshot shows a database playground interface. At the top, there's a query editor with the following SQL query:

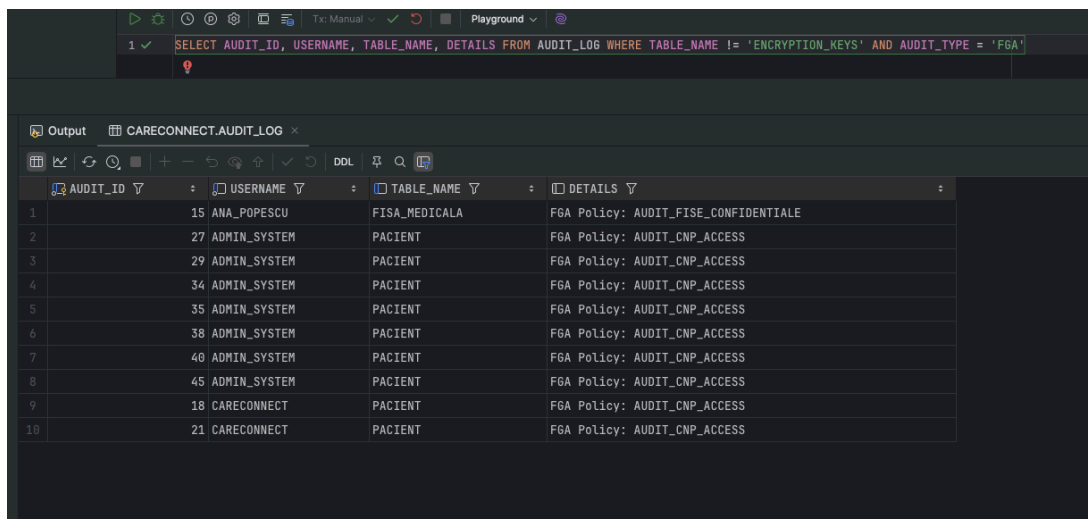
```
SELECT AUDIT_ID, USERNAME, TABLE_NAME, DETAILS FROM AUDIT_LOG WHERE TABLE_NAME != 'ENCRYPTION_KEYS' AND AUDIT_TYPE NOT IN ('DECRYPT', 'FGA')
```

Below the query editor, the results are displayed in a table with the following columns: AUDIT_ID, USERNAME, TABLE_NAME, and DETAILS. The table contains 24 rows of data.

AUDIT_ID	USERNAME	TABLE_NAME	DETAILS
1	16 ADMIN_SYSTEM	DBA_USERS	User creat: ANDREI, Rol: ROL_RECEPTIE, Grad: 0
2	17 ADMIN_SYSTEM	PERSONAL_MEDICAL	Angajat nou adaugat
3	30 ADMIN_SYSTEM	PACIENT	Date pacient actualizate
4	31 ADMIN_SYSTEM	PACIENT	Date pacient actualizate
5	32 ADMIN_SYSTEM	PACIENT	Date pacient actualizate
6	41 ADMIN_SYSTEM	PACIENT	Date pacient actualizate
7	42 ADMIN_SYSTEM	PACIENT	Date pacient actualizate
8	43 ADMIN_SYSTEM	PACIENT	Date pacient actualizate
9	1 SYS	DEPARTAMENT	Departament nou creat
10	2 SYS	DEPARTAMENT	Departament nou creat
11	3 SYS	DEPARTAMENT	Departament nou creat
12	4 SYS	PERSONAL_MEDICAL	Angajat nou adaugat
13	5 SYS	PERSONAL_MEDICAL	Angajat nou adaugat
14	6 SYS	PERSONAL_MEDICAL	Angajat nou adaugat
15	7 SYS	PERSONAL_MEDICAL	Angajat nou adaugat
16	8 SYS	PACIENT	Pacient nou inregistrat
17	9 SYS	PACIENT	Pacient nou inregistrat
18	10 SYS	PACIENT	Pacient nou inregistrat
19	11 SYS	FISA_MEDICALA	Fișă medicală nouă creată (nivel=1)
20	12 SYS	FISA_MEDICALA	Fișă medicală nouă creată (nivel=2)
21	13 SYS	FISA_MEDICALA	Fișă medicală nouă creată (nivel=3)
22	22 CARECONNECT	PACIENT	Date pacient actualizate
23	23 CARECONNECT	PACIENT	Date pacient actualizate
24	24 CARECONNECT	PACIENT	Date pacient actualizate

Figura 1.15: Audit prin triggers

2. **Fine-Grained Auditing (FGA):** Politici configurate pentru auditarea accesărilor la coloane sensibile (CNP, chei de criptare, fișe cu nivel confidențialitate 3)



The screenshot shows a database playground interface. At the top, there's a query editor with the following SQL query:

```
SELECT AUDIT_ID, USERNAME, TABLE_NAME, DETAILS FROM AUDIT_LOG WHERE TABLE_NAME != 'ENCRYPTION_KEYS' AND AUDIT_TYPE = 'FGA'
```

Below the query editor, the results are displayed in a table with the following columns: AUDIT_ID, USERNAME, TABLE_NAME, and DETAILS. The table contains 10 rows of data.

AUDIT_ID	USERNAME	TABLE_NAME	DETAILS
1	15 ANA_POPESCU	FISA_MEDICALA	FGA Policy: AUDIT_FISE_CONFIDENTIALE
2	27 ADMIN_SYSTEM	PACIENT	FGA Policy: AUDIT_CNP_ACCESS
3	29 ADMIN_SYSTEM	PACIENT	FGA Policy: AUDIT_CNP_ACCESS
4	34 ADMIN_SYSTEM	PACIENT	FGA Policy: AUDIT_CNP_ACCESS
5	35 ADMIN_SYSTEM	PACIENT	FGA Policy: AUDIT_CNP_ACCESS
6	38 ADMIN_SYSTEM	PACIENT	FGA Policy: AUDIT_CNP_ACCESS
7	40 ADMIN_SYSTEM	PACIENT	FGA Policy: AUDIT_CNP_ACCESS
8	45 ADMIN_SYSTEM	PACIENT	FGA Policy: AUDIT_CNP_ACCESS
9	18 CARECONNECT	PACIENT	FGA Policy: AUDIT_CNP_ACCESS
10	21 CARECONNECT	PACIENT	FGA Policy: AUDIT_CNP_ACCESS

Figura 1.16: Audit FGA

In aplicația CLI, doar cine are gradul 4 poate avea acces la audit logs:

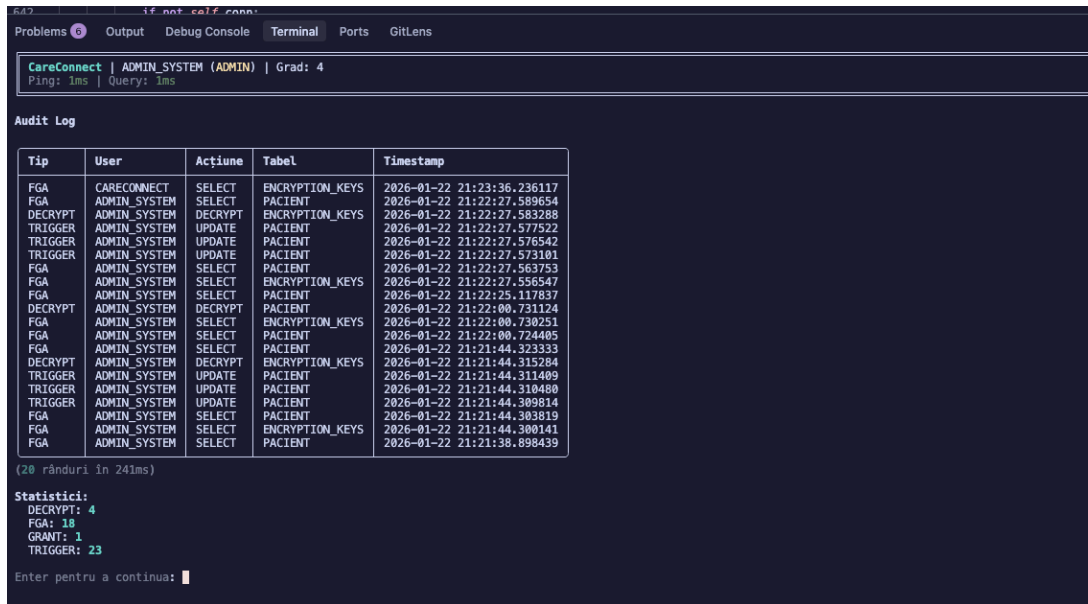


Figura 1.17: Audit - grad 4

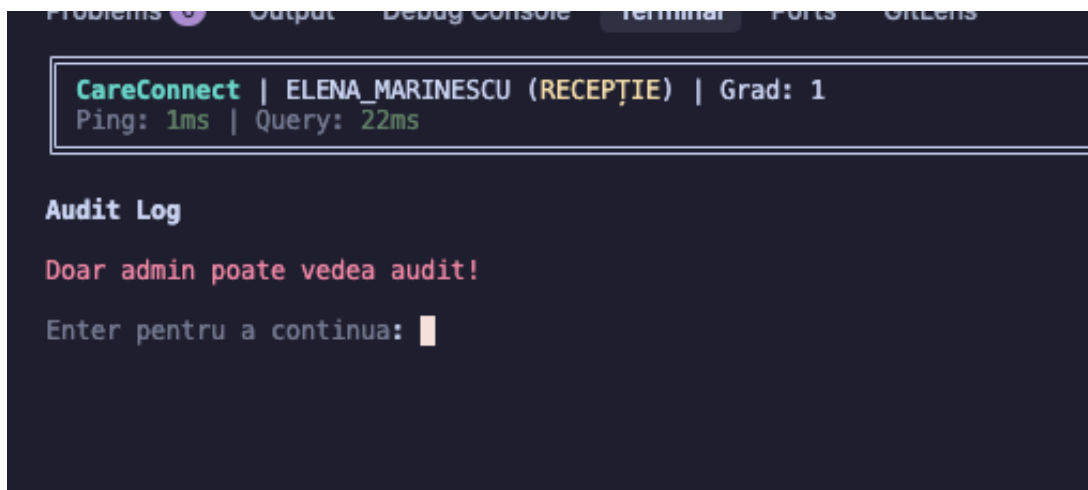


Figura 1.18: Audit - grad < 4

1.5.4 Prevenirea atacurilor

1. **SQL Injection:** Toate operațiile sunt realizate prin proceduri și funcții PL/SQL cu parametri bind, eliminând construirea dinamică de query-uri
2. **Application Context:** Context Oracle pentru stocarea gradului de acces în sesiune, setat automat la login
3. **Profiluri utilizatori:** Configurate pentru limitarea resurselor (CPU, sesiuni, timp idle) și gestionarea parolelor

1.5.5 Mascarea datelor

1. **Funcții de mascare:** Implementate pentru telefon, email, adresă și CNP (mascare parțială sau totală)
2. **Views per rol:** Fiecare rol are view-uri dedicate care aplică mascarea corespunzătoare nivelului de acces

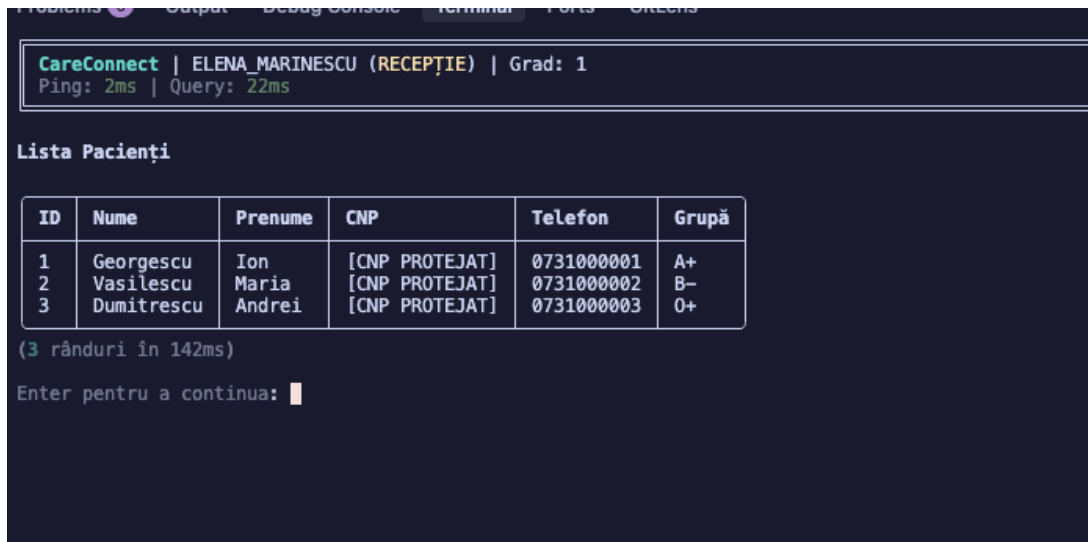


Figura 1.19: View pacienti - grad 1



Figura 1.20: View pacienti - grad 3

3. **Decriptare condiționată:** Doar medicii și adminii pot decrifica CNP-ul complet.



Figura 1.21: Decriptare CNP