

Sisteme de Gestiune a Bazelor de Date

Proiect

Sistem de Gestiune a Restaurării Patrimoniului Cultural

Student: Voaides-Negustor Robert-Ionuț

Grupa: 251

Seria: 25

Anul Universitar: 2025-2026

Cuprins

| | |
|--|-----------|
| 1 Descrierea Bazei de Date | 3 |
| 1.1 Scenariul Real Modelat | 3 |
| 1.2 Utilitatea Bazei de Date | 3 |
| 2 Diagrama Entitate-Relație (ERD) | 4 |
| 3 Diagrama Conceptuală | 5 |
| 4 Implementarea Structurii | 6 |
| 5 Popularea Bazei de Date | 9 |
| 6 Utilizarea Colecțiilor (Tablouri/Vectori) | 12 |
| 7 Utilizarea Cursoarelor | 13 |
| 8 Funcție Stocată | 14 |
| 9 Procedură Stocată | 15 |
| 10 Trigger LMD la Nivel de Comandă | 16 |
| 11 Trigger LMD la Nivel de Linie | 17 |
| 12 Trigger LDD | 18 |
| 13 Pachet PL/SQL (Optional) | 19 |

Introducere

Infrastructura utilizata

- **Sistem de operare:** Fedora Linux
- **Containerizare:** Docker
- **SGBD:** Oracle Database XE 21c (container `oracle-xe`, imagine `gvenzl/oracle-xe:21-slim`)
- **Mediu de lucru:** Visual Studio Code și DataGrip

1. Descrierea Bazei de Date

1.1 Scenariul Real Modelat

Această bază de date a fost proiectată pentru a gestiona activitatea unei instituții sau companii specializate în conservarea și restaurarea patrimoniului istoric. Scenariul real modelat urmărește ciclul de viață complet al unui proiect de restaurare, de la identificarea obiectivului (monumentul) și asigurarea surselor de finanțare, până la execuția efectivă și monitorizarea calității lucrărilor.

Baza de date surprinde interacțiunile complexe dintre diversele entități implicate într-un șantier de restaurare: monumentele istorice, echipele multidisciplinare de experți, resursele materiale necesare, instituțiile statului care avizează lucrările și sursele de finanțare.

1.2 Utilitatea Bazei de Date

Implementarea acestui sistem informatic răspunde nevoii de a avea o evidență clară, transparentă și centralizată asupra modului în care este conservat patrimoniul. Principalele funcționalități și beneficii sunt:

- **Trasabilitatea intervențiilor:** Sistemul permite crearea unui istoric detaliat al tuturor restaurărilor efectuate asupra unui monument. Se poate verifica oricând ce materiale s-au utilizat și ce experți au participat, informații vitale pentru viitoarele lucrări de întreținere.
- **Gestionarea resurselor:** Baza de date rezolvă problema alocării eficiente a resurselor. Deoarece experții și stocurile de materiale pot fi partajate între mai multe șantiere simultan, sistemul evidențiază clar distribuția acestora.
- **Monitorizarea finanțiară și legală:** Permite urmărirea bugetelor alocate din diverse surse de finanțare, precum și starea avizelor necesare de la autoritățile competente (ex: Ministerul Culturii, Primărie) și rezultatele inspecțiilor de șantier.

2. Diagrama Entitate-Relație (ERD)

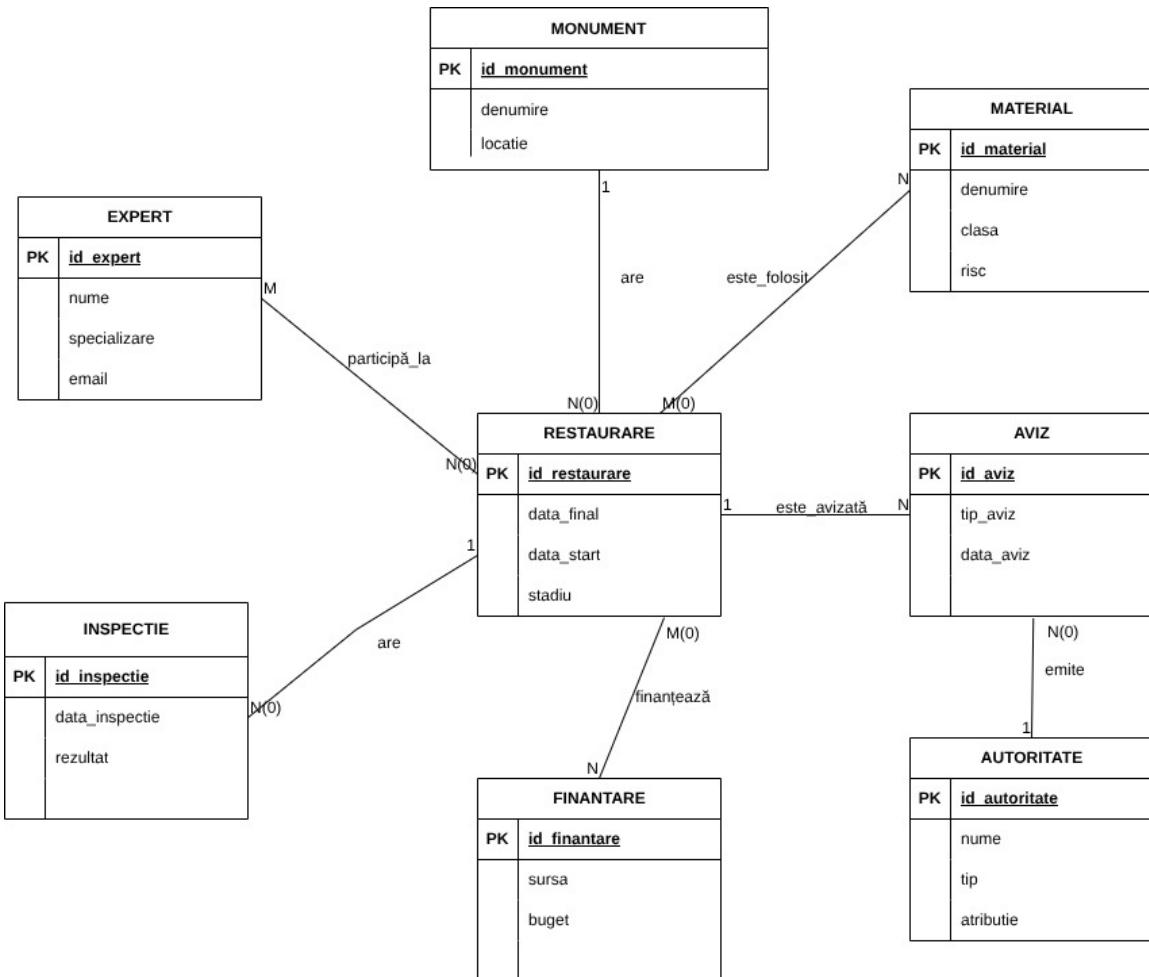


Figura 2.1: Diagrama ERD a bazei de date

3. Diagrama Conceptuală

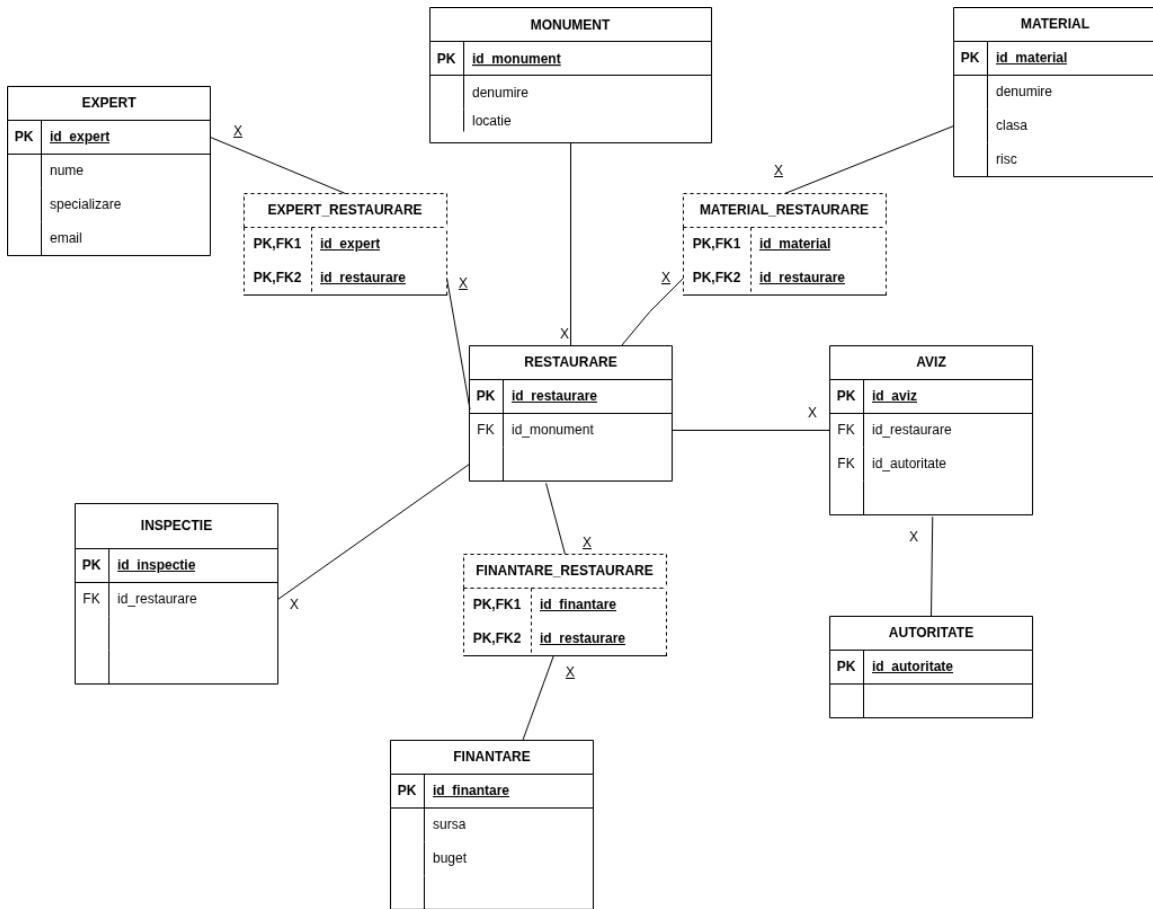


Figura 3.1: Diagrama Conceptuală

4. Implementarea Structurii

Mai jos sunt prezentate instrucțiunile de creare a tabelelor și constrângerile de integritate.

```
1 CREATE TABLE monument (
2     id_monument NUMBER(5) CONSTRAINT pk_monument PRIMARY KEY,
3     denumire     VARCHAR2(100) NOT NULL,
4     locatie      VARCHAR2(100) NOT NULL
5 );
6
7
8 CREATE TABLE expert (
9     id_expert      NUMBER(5) CONSTRAINT pk_expert PRIMARY KEY,
10    nume           VARCHAR2(50) NOT NULL,
11    specializare   VARCHAR2(50) NOT NULL,
12    email          VARCHAR2(100) CONSTRAINT uq_expert_email UNIQUE
13 );
14
15
16 CREATE TABLE material (
17     id_material    NUMBER(5) CONSTRAINT pk_material PRIMARY KEY,
18     denumire       VARCHAR2(50) NOT NULL,
19     clasa          VARCHAR2(20),
20     risc            VARCHAR2(20)
21 );
22
23
24 CREATE TABLE finantare (
25     id_finantare  NUMBER(5) CONSTRAINT pk_finantare PRIMARY KEY,
26     sursa          VARCHAR2(50) NOT NULL,
27     buget          NUMBER(12, 2) CONSTRAINT ck_buget_pozitiv CHECK (buget
> 0)
28 );
29
30
31 CREATE TABLE autoritate (
32     id_autoritate NUMBER(5) CONSTRAINT pk_autoritate PRIMARY KEY,
33     nume          VARCHAR2(100) NOT NULL,
34     tip            VARCHAR2(50),
35     atributie     VARCHAR2(100)
36 );
37
38 CREATE TABLE restaurare (
39     id_restaurare NUMBER(5) CONSTRAINT pk_restaurare PRIMARY KEY,
40     id_monument   NUMBER(5) NOT NULL,
41     data_start    DATE DEFAULT SYSDATE,
42     data_final   DATE,
43     stadiu        VARCHAR2(20) CONSTRAINT ck_stadiu CHECK (stadiu IN ('
Planificat', 'In executie', 'Finalizat', 'Suspendat')),
44     CONSTRAINT fk_restaurare_monument FOREIGN KEY (id_monument)
45     REFERENCES monument(id_monument),
46     CONSTRAINT ck_date_valide CHECK (data_final >= data_start)
47 );
48
49 CREATE TABLE aviz (
50     id_aviz         NUMBER(5) CONSTRAINT pk_aviz PRIMARY KEY,
51     id_restaurare  NUMBER(5) NOT NULL,
```

```

52     id_autoritate NUMBER(5) NOT NULL ,
53     tip_aviz      VARCHAR2(50) NOT NULL ,
54     data_aviz      DATE DEFAULT SYSDATE ,
55     CONSTRAINT fk_aviz_restaurare FOREIGN KEY (id_restaurare) REFERENCES
56     restaurare(id_restaurare),
57     CONSTRAINT fk_aviz_autoritate FOREIGN KEY (id_autoritate) REFERENCES
58     autoritate(id_autoritate)
59   );
60
61
62
63
64
65
66
67
68
69 CREATE TABLE inspectie (
70     id_inspectie    NUMBER(5) CONSTRAINT pk_inspectie PRIMARY KEY ,
71     id_restaurare  NUMBER(5) NOT NULL ,
72     data_inspectie DATE DEFAULT SYSDATE ,
73     rezultat        VARCHAR2(200) ,
74     CONSTRAINT fk_inspectie_restaurare FOREIGN KEY (id_restaurare)
75     REFERENCES restaurare(id_restaurare)
76   );
77
78
79
80
81
82
83
84
85
86
87 CREATE TABLE material_restaurare (
88     id_material     NUMBER(5) ,
89     id_restaurare  NUMBER(5) ,
90     CONSTRAINT pk_material_restaurare PRIMARY KEY (id_material ,
91     id_restaurare),
92     CONSTRAINT fk_mr_material FOREIGN KEY (id_material) REFERENCES
93     material(id_material),
94     CONSTRAINT fk_mr_restaurare FOREIGN KEY (id_restaurare) REFERENCES
95     restaurare(id_restaurare)
96   );
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193

```

Listing 4.1: Crearea tabelelor

Dovada rulării în Oracle:

```
Table MONUMENT created.

Table EXPERT created.

Table MATERIAL created.

Table FINANTARE created.

Table AUTORITATE created.

Table RESTAURARE created.

Table AVIZ created.

Table INSPECTIE created.

Table EXPERT_RESTAURARE created.

Table MATERIAL_RESTAURARE created.

Table FINANTARE_RESTAURARE created.
```

Figura 4.1: Rularea comenzilor CREATE TABLE

5. Popularea Bazei de Date

Au fost inserate minim 5 înregistrări pentru entitățile independente și 10 pentru cele asociative.

```
1 INSERT INTO monument VALUES (1, 'Castelul Peles', 'Sinaia');
2 INSERT INTO monument VALUES (2, 'Biserica Neagra', 'Brasov');
3 INSERT INTO monument VALUES (3, 'Cetatea de Scaun', 'Suceava');
4 INSERT INTO monument VALUES (4, 'Manastirea Voronet', 'Gura Humorului');
5 INSERT INTO monument VALUES (5, 'Cazinoul', 'Constanta');

6

7
8 INSERT INTO expert VALUES (101, 'Popescu Ion', 'Arhitect', 'popescu.
    i@expert.ro');
9 INSERT INTO expert VALUES (102, 'Ionescu Maria', 'Inginer Structurist',
    'maria.i@expert.ro');
10 INSERT INTO expert VALUES (103, 'Georgescu Vlad', 'Restaurator Pictura',
    'vlad.g@expert.ro');
11 INSERT INTO expert VALUES (104, 'Dumitru Ana', 'Arheolog', 'ana.d@expert
    .ro');
12 INSERT INTO expert VALUES (105, 'Stanescu Dan', 'Manager Proiect', 'dan.
    s@expert.ro');

13
14
15 INSERT INTO material VALUES (201, 'Piatra de rau', 'Natural', 'Scazut');
16 INSERT INTO material VALUES (202, 'Mortar hidraulic', 'Sintetic', 'Mediu
    ');
17 INSERT INTO material VALUES (203, 'Lemn de stejar tratat', 'Natural', 'Ridicat');
18 INSERT INTO material VALUES (204, 'Pigment mineral', 'Chimic', 'Mediu');
19 INSERT INTO material VALUES (205, 'Caramida arsa', 'Compozit', 'Scazut')
    ;

20
21
22 INSERT INTO finantare VALUES (301, 'Ministerul Culturii', 5000000);
23 INSERT INTO finantare VALUES (302, 'Fonduri Europene REGIO', 12000000);
24 INSERT INTO finantare VALUES (303, 'Buget Local Sinaia', 200000);
25 INSERT INTO finantare VALUES (304, 'Donatii Private ONG', 50000);
26 INSERT INTO finantare VALUES (305, 'Granturi Norvegiene', 3500000);

27
28
29 INSERT INTO autoritate VALUES (401, 'Ministerul Culturii', 'Guvernamental',
    'Avizare patrimoniu');
30 INSERT INTO autoritate VALUES (402, 'Primaria Sinaia', 'Local', 'Certificat Urbanism');
31 INSERT INTO autoritate VALUES (403, 'ISC Brasov', 'Inspectie', 'Control Calitate');
32 INSERT INTO autoritate VALUES (404, 'Directia Judeteana Cultura', 'Judetean',
    'Monitorizare');
33 INSERT INTO autoritate VALUES (405, 'Primaria Constanta', 'Local', 'Autorizatie Constructie');

34
35
36 INSERT INTO restaurare VALUES (1001, 1, TO_DATE('01-03-2024', 'DD-MM-YYYY
    '), TO_DATE('01-12-2025', 'DD-MM-YYYY'), 'In executie');
37 INSERT INTO restaurare VALUES (1002, 2, TO_DATE('15-05-2023', 'DD-MM-YYYY
    '), TO_DATE('15-05-2024', 'DD-MM-YYYY'), 'Finalizat');
```

```

38 INSERT INTO restaurare VALUES (1003, 5, TO_DATE('01-01-2020', 'DD-MM-YYYY'), TO_DATE('01-01-2026', 'DD-MM-YYYY'), 'In executie');
39 INSERT INTO restaurare VALUES (1004, 3, TO_DATE('10-10-2025', 'DD-MM-YYYY'), NULL, 'Planificat');
40 INSERT INTO restaurare VALUES (1005, 4, TO_DATE('01-06-2024', 'DD-MM-YYYY'), TO_DATE('01-09-2024', 'DD-MM-YYYY'), 'Suspendat');
41
42
43 INSERT INTO aviz VALUES (1, 1001, 401, 'Aviz Favorabil', TO_DATE('20-02-2024', 'DD-MM-YYYY'));
44 INSERT INTO aviz VALUES (2, 1001, 402, 'Autorizatie Constructie', TO_DATE('25-02-2024', 'DD-MM-YYYY'));
45 INSERT INTO aviz VALUES (3, 1003, 405, 'Prelungire Autorizatie', TO_DATE('10-01-2024', 'DD-MM-YYYY'));
46
47
48 INSERT INTO inspectie VALUES (1, 1001, TO_DATE('01-04-2024', 'DD-MM-YYYY'), 'Conform cu proiectul');
49 INSERT INTO inspectie VALUES (2, 1003, TO_DATE('15-04-2024', 'DD-MM-YYYY'), 'Degradari neprevazute la fatada');
50
51
52 INSERT INTO expert_restaureare VALUES (101, 1001);
53 INSERT INTO expert_restaureare VALUES (102, 1001);
54 INSERT INTO expert_restaureare VALUES (105, 1001);
55 INSERT INTO expert_restaureare VALUES (101, 1003);
56 INSERT INTO expert_restaureare VALUES (102, 1003);
57 INSERT INTO expert_restaureare VALUES (103, 1005);
58 INSERT INTO expert_restaureare VALUES (104, 1004);
59 INSERT INTO expert_restaureare VALUES (105, 1002);
60 INSERT INTO expert_restaureare VALUES (102, 1002);
61 INSERT INTO expert_restaureare VALUES (101, 1002);
62
63
64 INSERT INTO material_restaureare VALUES (203, 1001);
65 INSERT INTO material_restaureare VALUES (205, 1001);
66 INSERT INTO material_restaureare VALUES (201, 1002);
67 INSERT INTO material_restaureare VALUES (202, 1002);
68 INSERT INTO material_restaureare VALUES (201, 1003);
69 INSERT INTO material_restaureare VALUES (202, 1003);
70 INSERT INTO material_restaureare VALUES (204, 1003);
71 INSERT INTO material_restaureare VALUES (201, 1004);
72 INSERT INTO material_restaureare VALUES (204, 1005);
73 INSERT INTO material_restaureare VALUES (202, 1005);
74
75
76 INSERT INTO finantare_restaureare VALUES (301, 1001);
77 INSERT INTO finantare_restaureare VALUES (303, 1001);
78 INSERT INTO finantare_restaureare VALUES (304, 1002);
79 INSERT INTO finantare_restaureare VALUES (305, 1002);
80 INSERT INTO finantare_restaureare VALUES (302, 1003);
81 INSERT INTO finantare_restaureare VALUES (301, 1003);
82 INSERT INTO finantare_restaureare VALUES (305, 1003);
83 INSERT INTO finantare_restaureare VALUES (302, 1004);
84 INSERT INTO finantare_restaureare VALUES (301, 1005);
85 INSERT INTO finantare_restaureare VALUES (304, 1005);
86
87

```

```
88 COMMIT;
```

Listing 5.1: Inserarea datelor

Dovada rulării în Oracle:

Figura 5.1: Rezultatul inserării datelor

6. Utilizarea Colecțiilor (Tablouri/-Vectori)

Enunțul Problemei

[Formulați problema în limbaj natural aici...]

Rezolvare PL/SQL

```
1 CREATE OR REPLACE PROCEDURE demo_colectii IS
2     TYPE t_tablou IS TABLE OF VARCHAR2(100);
3     v_lista t_tablou := t_tablou();
4 BEGIN
5     -- Codul tau aici
6     NULL;
7 END;
8 /
```

Listing 6.1: Subprogram cu colecții

Apel și Rezultat

```
1 BEGIN
2     demo_colectii;
3 END;
4 /
```

Figura 6.1: Execuția cerinței 6

7. Utilizarea Cursoarelor

Enunțul Problemei

[Problema care necesită 2 tipuri de cursoare, unul parametrizat...]

Rezolvare PL/SQL

```
1 -- Codul tau aici
```

Listing 7.1: Subprogram cu cursoare

Apel și Rezultat

Figura 7.1: Execuția cerinței 7

8. Funcție Stocată

Enunțul Problemei

[Funcție care utilizează 3 tabele și tratează exceptiile...]

Rezolvare PL/SQL

```
1 -- Codul tau aici
```

Listing 8.1: Funcția PL/SQL

Apel și Rezultat (Tratare Exceptii)

Figura 8.1: Apelul funcției și tratarea exceptiilor

9. Procedură Stocată

Enunțul Problemei

[Procedură cu 5 tabele și excepții proprii...]

Rezolvare PL/SQL

```
1 -- Codul tau aici
```

Listing 9.1: Procedura PL/SQL

Apel și Rezultat

Figura 9.1: Execuția procedurii

10. Trigger LMD la Nivel de Comandă

Definire Trigger

```
1 -- Codul trigger-ului
```

Declansare

Figura 10.1: Demonstrație declansare trigger

11. Trigger LMD la Nivel de Linie

Definire Trigger

```
1 -- Codul trigger-ului
```

Declansare

Figura 11.1: Demonstrație declansare trigger

12. Trigger LDD

Definire Trigger

```
1 -- Codul trigger-ului pentru comenzi de tip CREATE/DROP/ALTER
```

Declansare

Figura 12.1: Demonstrație declansare trigger LDD

13. Pachet PL/SQL (Optional)

Specificația Pachetului

```
1 -- Codul specificatiei
```

Corpul Pachetului

```
1 -- Codul body-ului
```

Apel Funcționalități

Figura 13.1: Execuția elementelor din pachet