

# Raport Proiect Baze de Date

Ivan Vlad-Daniel, 331AB

Insert, Update, Delete:

## PolitistRepository:

```
26      // INSERT NATIV
27      @Modifying 1usage new *
28      @Transactional
29      @Query(value = "" +
30          "INSERT INTO Politisti (nume, prenume, grad, functie, telefon_serviciu) " +
31          "VALUES (:nume, :prenume, :grad, :functie, :telefon)",
32          nativeQuery = true)
33      void adaugaPolitistManual(
34          @Param("nume") String nume,
35          @Param("prenume") String prenume,
36          @Param("grad") String grad,
37          @Param("functie") String functie,
38          @Param("telefon") String telefon
39      );
40
41
42      // UPDATE NATIV
43      @Modifying 1usage new *
44      @Transactional
45      @Query(value = "" +
46          "UPDATE Politisti " +
47          "SET nume = :nume, " +
48          "    prenume = :prenume, " +
49          "    grad = :grad, " +
50          "    functie = :functie, " +
51          "    telefon_serviciu = :telefon_serviciu " +
52          "WHERE id_politist = :id_politist",
53          nativeQuery = true)
54      void updatePolitist(
55          @Param("id_politist") Integer id_politist,
56          @Param("nume") String nume,
57          @Param("prenume") String prenume,
58
59      // DELETE NATIV
60      @Modifying 1usage new *
61      @Transactional
62      @Query(value = "DELETE FROM Politisti WHERE id_politist = :id", nativeQuery = true)
63      void stergePolitistManual(@Param("id") Integer id);
```

PersoanaRepository:

```
// INSERT (Parametri Variabili: Date Personale)
@Modifying 1 usage new *
@Transactional
@Query(value = "" + 
    "INSERT INTO Persoane (nume, prenume, cnp, data_nasterii, telefon) " +
    "VALUES (:nume, :prenume, :cnp, :dataNasterii, :telefon)",
    nativeQuery = true)
void insertPersoana(
    @Param("nume") String nume,
    @Param("prenume") String prenume,
    @Param("cnp") String cnp,
    @Param("dataNasterii") LocalDate dataNasterii,
    @Param("telefon") String telefon
);
```

```

46     // UPDATE (Parametri Variabili: ID + Date Noi)
47     @Modifying 1 usage new *
48     @Transactional
49     @Query(value = "" +
50         "UPDATE Persoane " +
51         "SET nume = :nume, " +
52         "  prenume = :prenume, " +
53         "  cnp = :cnp, " +
54         "  data_nasterii = :dataNasterii, " +
55         "  telefon = :telefon " +
56         "WHERE id_persoana = :id",
57         nativeQuery = true)
58     void updatePersoana(
59         @Param("id") Integer id,
60         @Param("nume") String nume,
61         @Param("prenume") String prenume,
62         @Param("cnp") String cnp,
63         @Param("dataNasterii") LocalDate dataNasterii,
64         @Param("telefon") String telefon
65     );
66
67     // DELETE (Parametru Variabil: ID)
68     @Modifying 1 usage & vladiv5
69     @Transactional
70     @Query(value = "DELETE FROM Persoane WHERE id_persoana = :id", nativeQuery = true)
71     void deletePersoanaNative(@Param("id") Integer id);
72

```

IncidentRepository:

```
29     // INSERT (Parametri Variabili)
30     @Modifying 1 usage new *
31     @Transactional
32     @Query(value = "" +
33             "INSERT INTO Incidente (tip_incident, data_emitere, descriere_locatie, " +
34             "descriere_incident, id_politist_responsabil, " +
35             "id_adresa_incident, status) " +
36             "VALUES (:tip, :data, :locatie, :descriere, :idPolitist, :idAdresa, :status)",
37             nativeQuery = true)
38     void insertIncident(
39         @Param("tip") String tip,
40         @Param("data") LocalDateTime data,
41         @Param("locatie") String locatie,
42         @Param("descriere") String descriere,
43         @Param("idPolitist") Integer idPolitist,
44         @Param("idAdresa") Integer idAdresa,
45         @Param("status") String status
46     );
```

```

    ...
48     // UPDATE (Parametri Variabili)
49     @Modifying 1usage new *
50     @Transactional
51     @Query(value = "" +
52             "UPDATE Incidente " +
53             "SET tip_incident = :tip, " +
54             "    data_emitere = :data, " +
55             "    descriere_locatie = :locatie, " +
56             "    descriere_incident = :descriere, " +
57             "    id_politist_responsabil = :idPolitist, " +
58             "    id_adresa_incident = :idAdresa, " +
59             "    status = :status " +
60             "WHERE id_incident = :id",
61             nativeQuery = true)
62     void updateIncident(
63             @Param("id") Integer id,
64             @Param("tip") String tip,
65             @Param("data") LocalDateTime data,
66             @Param("locatie") String locatie,
67             @Param("descriere") String descriere,
68             @Param("idPolitist") Integer idPolitist,
69             @Param("idAdresa") Integer idAdresa,
70             @Param("status") String status
71     );
72
73     // DELETE (Parametru Variabil: ID)
74     @Modifying 1usage & vladiv5
75     @Transactional
76     @Query(value = "DELETE FROM Incidente WHERE id_incident = :id", nativeQuery = true)
77     void deleteIncidentNative(@Param("id") Integer id);
78
79

```

AmendaRepository:

```
30     // INSERT (Parametri Variabili: Detalii Amenda)
31     @Modifying 1 usage new *
32     @Transactional
33     @Query(value = "" +
34         "INSERT INTO Amenzi (motiv, suma, stare_plata, data_emitere, id_politist, id_persoana) " +
35         "VALUES (:motiv, :suma, :stare, :data, :idPolitist, :idPersoana)",
36         nativeQuery = true)
37     void insertAmenda(
38         @Param("motiv") String motiv,
39         @Param("suma") BigDecimal suma,
40         @Param("stare") String stare,
41         @Param("data") LocalDateTime data,
42         @Param("idPolitist") Integer idPolitist,
43         @Param("idPersoana") Integer idPersoana
44     );
45
46     // UPDATE (Parametri Variabili: ID + Date Noi)
47     @Modifying 1 usage new *
48     @Transactional
49     @Query(value = "" +
50         "UPDATE Amenzi " +
51         "SET motiv = :motiv, " +
52         "    suma = :suma, " +
53         "    stare_plata = :stare, " +
54         "    data_emitere = :data, " +
55         "    id_politist = :idPolitist, " +
56         "    id_persoana = :idPersoana " +
57         "WHERE id_amenda = :id",
58         nativeQuery = true)
59     void updateAmenda(
60         @Param("id") Integer id,
61         @Param("motiv") String motiv,
62         @Param("suma") BigDecimal suma,
63         @Param("stare") String stare,
64         @Param("data") LocalDateTime data,
65         @Param("idPolitist") Integer idPolitist,
66         @Param("idPersoana") Integer idPersoana
67     );
68
```

```
69 // DELETE (Parametru Variabil: ID)
70 @Modifying 1 usage  à vladiv5
71 @Transactional
72 @Query(value = "DELETE FROM Amenzi WHERE id_amenda = :id", nativeQuery = true)
73 void deleteAmendaNative(@Param("id") Integer id);
74
```

AdresaRepository:

```
28     // INSERT (Parametri Variabili: Detalii Adresa)
29     @Modifying 1usage new *
30     @Transactional
31     @Query(value = "" +
32         "INSERT INTO Adresa (judet_sau_sector, localitate, strada, numar, bloc, apartament) " +
33         "VALUES (:judet, :localitate, :strada, :numar, :bloc, :ap)",
34         nativeQuery = true)
35     void insertAdresa(
36         @Param("judet") String judet,
37         @Param("localitate") String localitate,
38         @Param("strada") String strada,
39         @Param("numar") String numar,
40         @Param("bloc") String bloc,
41         @Param("ap") Integer ap
42     );
43
44     // UPDATE (Parametri Variabili: ID + Date Noi)
45     @Modifying 1usage new *
46     @Transactional
47     @Query(value = "" +
48         "UPDATE Adresa " +
49         "SET judet_sau_sector = :judet, " +
50         "    localitate = :localitate, " +
51         "    strada = :strada, " +
52         "    numar = :numar, " +
53         "    bloc = :bloc, " +
54         "    apartament = :ap " +
55         "WHERE id_adresa = :id",
56         nativeQuery = true)
57     void updateAdresa(
58         @Param("id") Integer id,
59         @Param("judet") String judet,
60         @Param("localitate") String localitate,
61         @Param("strada") String strada,
62         @Param("numar") String numar,
63         @Param("bloc") String bloc,
64         @Param("ap") Integer ap
65     );
66
```

```

~~
67     // DELETE (Parametru Variabil: ID)
68     @Modifying 1 usage  vladiv5
69     @Transactional
70     @Query(value = "DELETE FROM Adresa WHERE id_adresa = :id", nativeQuery = true)
71     void deleteAdresaNative(@Param("id") Integer id);
72

```

Interogari simple:

PolitistRepository:

```

76     // Top Politisti dupa valoarea amenzilor (JOIN + SUM + GROUP BY)
77     // [Parametru Variabil]: Intervalul de date (startDate, endDate)
78     @Query(value = "" + 1 usage new *
79         "SELECT p.nume, p.prenume, p.grad, p.functie, SUM(a.suma) as total_valoare " +
80         "FROM politisti p " +
81         "JOIN amenzi a ON p.id_politist = a.id_politist " +
82         "WHERE (:startDate IS NULL OR a.data_emitere >= :startDate) " +
83         "AND (:endDate IS NULL OR a.data_emitere <= :endDate) " +
84         "GROUP BY p.id_politist, p.nume, p.prenume, p.grad, p.functie " +
85         "ORDER BY total_valoare DESC",
86         nativeQuery = true)
87     List<Map<String, Object>> getTopPolitistiAmenzi(
88         @Param("startDate") LocalDateTime startDate,
89         @Param("endDate") LocalDateTime endDate
90     );
91
92
93     // Statistici grupate pe Grade (JOIN + COUNT + SUM)
94     // [Parametru Variabil]: Intervalul de date
95     @Query(value = "" + 1 usage new *
96         "SELECT p.grad, " +
97             "COUNT(a.id_amenda) as nr_amenzi, " +
98             "SUM(a.suma) as valoare_totala " +
99             "FROM politisti p " +
100            "JOIN amenzi a ON p.id_politist = a.id_politist " +
101            "WHERE (:startDate IS NULL OR a.data_emitere >= :startDate) " +
102            "AND (:endDate IS NULL OR a.data_emitere <= :endDate) " +
103            "GROUP BY p.grad",
104            nativeQuery = true)
105    List<Map<String, Object>> getStatisticiPerGrad(
106        @Param("startDate") LocalDateTime startDate,
107        @Param("endDate") LocalDateTime endDate
108    );

```

PersoanaRepository:

```
78     // Raport "Rau Platnici": Persoane cu datorii neplatite (JOIN Persoane-Amenzi)
79     // [Parametri Variabili]: Intervalul de date
80     @Query(value = "" + 1usage new *
81         "SELECT p.nume, p.prenume, p.cnp, SUM(a.suma) as datorie_totala " +
82         "FROM persoane p " +
83         "JOIN amenzi a ON p.id_persoana = a.id_persoana " +
84         "WHERE a.stare_plata = 'Neplatita' " +
85         "AND (:startDate IS NULL OR a.data_emitere >= :startDate) " +
86         "AND (:endDate IS NULL OR a.data_emitere <= :endDate) " +
87         "GROUP BY p.id_persoana, p.nume, p.prenume, p.cnp " +
88         "ORDER BY datorie_totala DESC",
89         nativeQuery = true)
90     List<Map<String, Object>> getRauPlatnici(
91         @Param("startDate") LocalDateTime startDate,
92         @Param("endDate") LocalDateTime endDate
93     );
```

IncidentRepository:

```

84     // Raport "Dosar Politist": Istoricul cazurilor lucrate (JOIN Incidente-Politisti-Adrese)
85     // [Parametri Variabili]: ID Politist + Interval Data
86     @Query(value = "" + 1usage & vladiv5
87             +
88             "SELECT i.tip_incident, i.data_emitere, i.descriere_locatie, adr.strada " +
89             "FROM incidente i " +
90             "JOIN politisti p ON i.id_politist_responsabil = p.id_politist " +
91             "JOIN adrese adr ON i.id_adresa_incident = adr.id_adresa " +
92             "WHERE p.id_politist = :idPolitist " +
93             "AND (:startDate IS NULL OR i.data_emitere >= :startDate) " +
94             "AND (:endDate IS NULL OR i.data_emitere <= :endDate)",
95             nativeQuery = true)
96     List<Map<String, Object>> getIncidenteByPolitist(
97         @Param("idPolitist") Integer idPolitist,
98         @Param("startDate") LocalDateTime startDate,
99         @Param("endDate") LocalDateTime endDate
100    );
101
102
103
104     // Raport "Top Strazi Periculoase": Agregare dupa locatie
105     // [Complexitate]: JOIN + GROUP BY + COUNT + ORDER BY
106     @Query(value = "" + 1usage new *
107             +
108             "SELECT adr.strada, adr.localitate, COUNT(i.id_incident) as nr_incidente " +
109             "FROM adrese adr " +
110             "JOIN incidente i ON adr.id_adresa = i.id_adresa_incident " +
111             "WHERE (:startDate IS NULL OR i.data_emitere >= :startDate) " +
112             "AND (:endDate IS NULL OR i.data_emitere <= :endDate) " +
113             "GROUP BY adr.strada, adr.localitate " +
114             "ORDER BY nr_incidente DESC",
115             nativeQuery = true)
116     List<Map<String, Object>> getTopStraziIncidente(
117         @Param("startDate") LocalDateTime startDate,
118         @Param("endDate") LocalDateTime endDate
119    );

```

AmendaRepository:

```

80  // Raport "Cazier Fiscal": Istorici amenzi per CNP (JOIN Triplu: Amenzi-Persoane-Politisti)
81  // [Parametri Variabili]: CNP + Interval Data
82  @Query(value = "" + 1 usage new *
83      "SELECT a.motiv, a.suma, a.data_emitere, a.stare_plata, " +
84      "      pol.nume as nume_politist, pol.prenume as prenume_politist " +
85      "FROM amenzi a " +
86      "JOIN persoane p ON a.id_persoana = p.id_persoana " +
87      "JOIN politisti pol ON a.id_politist = pol.id_politist " +
88      "WHERE p.cnp = :cnp " +
89      "  AND (:startDate IS NULL OR a.data_emitere >= :startDate) " +
90      "  AND (:endDate IS NULL OR a.data_emitere <= :endDate)",
91      nativeQuery = true)
92  List<Map<String, Object>> getAmenziByCNP(
93      @Param("cnp") String cnp,
94      @Param("startDate") LocalDateTime startDate,
95      @Param("endDate") LocalDateTime endDate
96  );

```

AdresaRepository:

Interogari complexe:

## PolitistRepository:

```
113 // Agenti Severi: Politistii cu media amenzilor peste media globala a sectiei  
114 // [Complexitate]: Subquery in HAVING + AVG  
115 // [Parametru Variabil]: Intervalul de date  
116 @Query(value = "" + 1 usage new *  
117     "SELECT p.nume, p.prenume, p.grad, p.functie, AVG(a.suma) as medie_personala " +  
118     "FROM Politisti p " +  
119     "JOIN Amenzi a ON p.id_politist = a.id_politist " +  
120     "WHERE (:startDate IS NULL OR a.data_emitere >= :startDate) " +  
121     " AND (:endDate IS NULL OR a.data_emitere <= :endDate) " +  
122     "GROUP BY p.id_politist, p.nume, p.prenume, p.grad, p.functie " +  
123     "HAVING AVG(a.suma) > (" +  
124         "    SELECT AVG(a2.suma) " +  
125         "    FROM Amenzi a2 " +  
126         "    WHERE (:startDate IS NULL OR a2.data_emitere >= :startDate) " +  
127         "    AND (:endDate IS NULL OR a2.data_emitere <= :endDate)" +  
128     ") " +  
129     "ORDER BY medie_personala DESC",  
130     nativeQuery = true)  
131 List<Map<String, Object>> getAgentiSeveri(  
132     @Param("startDate") LocalDateTime startDate,  
133     @Param("endDate") LocalDateTime endDate  
134 );
```

### PersoanaRepository:

```
...
100   ...
101   ...
102   ...
103   ...
104   ...
105   ...
106   ...
107   ...
108   ...
109   ...
110   ...
111   ...
112   ...
113   ...
114   ...
115   ...
116   ...
117   ...
118   ...
119   ...
120   ...
121   ...;
```

### IncidentRepository:

```

120  // Raport "Zile Critice": Zile cu criminalitate peste media zilnică globală
121  // [Complexitate]: Subquery in HAVING + AVG + GROUP BY DATE()
122  @Query(value = "" + 1 usage  & vladiv5
123      "SELECT CAST(i.data_emitere AS DATE) as ziua, COUNT(*) as nr_incidente " +
124      "FROM Incidente i " +
125      "WHERE (:startDate IS NULL OR i.data_emitere >= :startDate) " +
126      " AND (:endDate IS NULL OR i.data_emitere <= :endDate) " +
127      "GROUP BY CAST(i.data_emitere AS DATE) " +
128      "HAVING COUNT(*) > (" +
129          "    SELECT AVG(sub.zilnic) FROM (" +
130              "        SELECT COUNT(*) as zilnic " +
131              "        FROM Incidente i2 " +
132              "        WHERE (:startDate IS NULL OR i2.data_emitere >= :startDate) " +
133              "        AND (:endDate IS NULL OR i2.data_emitere <= :endDate) " +
134              "        GROUP BY CAST(i2.data_emitere AS DATE)" +
135              "    ) as sub" +
136          ") " +
137          "ORDER BY nr_incidente DESC",
138          nativeQuery = true)
139  List<Map<String, Object>> getZileCritice(
140      @Param("startDate") LocalDateTime startDate,
141      @Param("endDate") LocalDateTime endDate
142  );

```

```

160  // 3. Zone Sigure (Complexă cu Subcerere)
161  // Cerință: Strazi unde nu s-au înregistrat incidente
162  // Implementare: Selectam strazile a căror ID nu se află în lista ID-urilor cu incidente (Subquery)
163  @Query(value = "" + 1 usage  new *
164      "SELECT a.strada, a.localitate " +
165      "FROM Adrese a " +
166      "WHERE a.id_adresa NOT IN (" +
167          "    SELECT i.id_adresa_incident " +
168          "    FROM Incidente i " +
169          "    WHERE i.id_adresa_incident IS NOT NULL " +
170          "        AND (:startDate IS NULL OR i.data_emitere >= :startDate) " +
171          "        AND (:endDate IS NULL OR i.data_emitere <= :endDate)" +
172      ") " +
173      "GROUP BY a.strada, a.localitate " +
174      "ORDER BY a.localitate, a.strada",
175      nativeQuery = true)
176  List<Map<String, Object>> getZoneSigure(
177      @Param("startDate") LocalDateTime startDate,
178      @Param("endDate") LocalDateTime endDate
179  );

```

AmendaRepository:

AdresaRepository:

Bonus:

### PolitistRepository:

```
141      // Search Intelligent (CONCAT + COLLATE)
142      // [Parametru Variabil]: Termenul de cautare
143      @Query(value = "" + 1 usage & vladiv5
144          "SELECT * FROM Politisti WHERE " +
145          "CONCAT(COALESCE(nume, ''), ' ', COALESCE(prenume, ''), ' ', " +
146          "COALESCE(grad, ''), ' ', COALESCE(functie, ''), ' ', " +
147          "COALESCE(telefon_serviciu, '')) " +
148          "COLLATE Latin1_General_100_CI_AI " +
149          "LIKE CONCAT(:termen, '%')",
150          nativeQuery = true)
151      List<Politist> cautaDupaInceput(@Param("termen") String termen);
```

### PersoanaRepository:

```
128      // Search Intelligent (CONCAT + COLLATE pentru diacritice)
129      @Query(value = "" + 1 usage & vladiv5
130          "SELECT * FROM Persoane WHERE " +
131          "CONCAT(COALESCE(nume, ''), ' ', COALESCE(prenume, ''), ' ', " +
132          "COALESCE(cnp, ''), ' ', COALESCE(telefon, '')) " +
133          "COLLATE Latin1_General_100_CI_AI " +
134          "LIKE CONCAT(:termen, '%')",
135          nativeQuery = true)
136      List<Persoana> cautaDupaInceput(@Param("termen") String termen);
137
138      // Valideaza Unicitate CNP (COUNT conditionat)
139      @Query(value = "" + 2 usages & vladiv5
140          "SELECT count(*) FROM Persoane " +
141          "WHERE cnp = :cnp " +
142          "AND (:idExclus IS NULL OR id_persoana != :idExclus)",
143          nativeQuery = true)
144      int verificaCnpUnic(@Param("cnp") String cnp, @Param("idExclus") Integer idExclus);
145
146      // Valideaza Unicitate Telefon
147      @Query(value = "" + 2 usages & vladiv5
148          "SELECT count(*) FROM Persoane " +
149          "WHERE telefon = :tel " +
150          "AND (:idExclus IS NULL OR id_persoana != :idExclus)",
151          nativeQuery = true)
152      int verificaTelefonUnic(@Param("tel") String tel, @Param("idExclus") Integer idExclus);
```

### IncidentRepository:

```

L00
L01    // Live Search: Cauta in Incidente SI in numele Politistului (LEFT JOIN)
L02    // [Parametru Variabil]: Termenul de cautare
L03    @Query(value = "" + 1 usage new *
L04        "SELECT i.* FROM Incidente i " +
L05        "LEFT JOIN Politisti p ON i.id_politist_responsabil = p.id_politist " +
L06        "WHERE " +
L07        "CONCAT(COALESCE(i.tip_incident, ''), ' ', COALESCE(i.descriere_locatie, ''), ' ', " +
L08        "COALESCE(i.descriere_incident, ''), ' ', COALESCE(p.nume, ''), ' ', " +
L09        "COALESCE(p.prenume, '')) " +
L10        "COLLATE Latin1_General_100_CI_AI " +
L11        "LIKE CONCAT(:termen, '%')",
L12        nativeQuery = true)
L13    List<Incident> cautaDupaInceput(@Param("termen") String termen);

```

### AmendaRepository:

```

98    // Live Search Avansat (Search prin JOIN-uri)
99    // Cauta in motivul amenzii, dar si in numele persoanei sau politistului
100   @Query(value = "" + 1 usage & vladiv5
101       "SELECT a.* FROM Amenzi a " +
102       "LEFT JOIN Persoane p ON a.id_persoana = p.id_persoana " +
103       "LEFT JOIN Politisti pol ON a.id_politist = pol.id_politist " +
104       "WHERE " +
105       "CONCAT(COALESCE(a.motiv, ''), ' ', COALESCE(p.nume, ''), ' ', " +
106       "COALESCE(p.prenume, ''), ' ', COALESCE(pol.nume, ''), ' ', " +
107       "COALESCE(pol.prenume, '')) " +
108       "COLLATE Latin1_General_100_CI_AI " +
109       "LIKE CONCAT(:termen, '%')",
110       nativeQuery = true)
111   List<Amenda> cautaDupaInceput(@Param("termen") String termen);
112

```

### AdresaRepository:

```

78    // Search Intelligent: Cauta in toate campurile adresei simultan
79    // Utilizeaza COALESCE pentru a gestiona campurile NULL (ex: bloc/ap)
80    @Query(value = "" + 1 usage & vladiv5
81        "SELECT * FROM Adrese WHERE " +
82        "CONCAT(COALESCE(strada, ''), ' ', COALESCE(numar, ''), ' ', " +
83        "COALESCE(bloc, ''), ' ', COALESCE(localitate, ''), ' ', " +
84        "COALESCE(judet_sau_sector, '')) " +
85        "COLLATE Latin1_General_100_CI_AI " +
86        "LIKE CONCAT(:termen, '%')",
87        nativeQuery = true)
88    List<Adresa> cautaDupaInceput(@Param("termen") String termen);
89

```