

# Raport Proiect Baze de Date

Ivan Vlad-Daniel, 331AB

Insert, Update, Delete:

## PolitistRepository:

```
26
27 // INSERT NATIV
28 @Modifying 1 usage new *
29 @Transactional
30 @Query(value = "" +
31     "INSERT INTO Politisti (nume, prenume, grad, functie, telefon_serviciu) " +
32     "VALUES (:nume, :prenume, :grad, :functie, :telefon)",
33     nativeQuery = true)
34 void adaugaPolitistManual(
35     @Param("nume") String nume,
36     @Param("prenume") String prenume,
37     @Param("grad") String grad,
38     @Param("functie") String functie,
39     @Param("telefon") String telefon
40 );
41
42 // UPDATE NATIV
43 @Modifying 1 usage 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 1 usage 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 1usage 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 3 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 1usage 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 1 usage 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 1 usage 3 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 );

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 );

```

```
69 // DELETE (Parametru Variabil: ID)
70 @Modifying 1 usage 3 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 Adrese (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 );

44 // UPDATE (Parametri Variabili: ID + Date Noi)
45 @Modifying 1usage new *
46 @Transactional
47 @Query(value = "" +
48     "UPDATE Adrese " +
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 );

```

```

67 // DELETE (Parametru Variabil: ID)
68 @Modifying 1 usage 3 vladiv5
69 @Transactional
70 @Query(value = "DELETE FROM Adrese 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
2 // Statistici grupate pe Grade (JOIN + COUNT + SUM)
3 // [Parametru Variabil]: Intervalul de date
4 @Query(value = "" + 1 usage new *
5     "SELECT p.grad, " +
6     "    COUNT(a.id_amenda) as nr_amenzi, " +
7     "    SUM(a.suma) as valoare_totala " +
8     "FROM politisti p " +
9     "JOIN amenzi a ON p.id_politist = a.id_politist " +
10    "WHERE (:startDate IS NULL OR a.data_emitere >= :startDate) " +
11    " AND (:endDate IS NULL OR a.data_emitere <= :endDate) " +
12    "GROUP BY p.grad",
13    nativeQuery = true)
14 List<Map<String, Object>> getStatisticiPerGrad(
15     @Param("startDate") LocalDateTime startDate,
16     @Param("endDate") LocalDateTime endDate
17 );

```

PersoanaRepository:

```

78 // Raport "Rau Platnici": Persoane cu datorii neplatite (JOIN Persoane-Amenzi)
79 // [Parametri Variabili]: Intervalul de date
80 @Query(value = "" + 1 usage 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 = "" + 1 usage 3 vladiv5
87 "SELECT i.tip_incident, i.data_emitere, i.descriere_locatie, adr.strada " +
88 "FROM incidente i " +
89 "JOIN politisti p ON i.id_politist_responsabil = p.id_politist " +
90 "JOIN adrese adr ON i.id_adresa_incident = adr.id_adresa " +
91 "WHERE p.id_politist = :idPolitist " +
92 " AND (:startDate IS NULL OR i.data_emitere >= :startDate) " +
93 " AND (:endDate IS NULL OR i.data_emitere <= :endDate)",
94 nativeQuery = true)
95 List<Map<String, Object>> getIncidenteByPolitist(
96     @Param("idPolitist") Integer idPolitist,
97     @Param("startDate") LocalDateTime startDate,
98     @Param("endDate") LocalDateTime endDate
99 );
143
144 // Raport "Top Strazi Periculoase": Agregare dupa locatie
145 // [Complexitate]: JOIN + GROUP BY + COUNT + ORDER BY
146 @Query(value = "" + 1 usage new *
147 "SELECT adr.strada, adr.localitate, COUNT(i.id_incident) as nr_incidente "
148 "FROM adrese adr " +
149 "JOIN incidente i ON adr.id_adresa = i.id_adresa_incident " +
150 "WHERE (:startDate IS NULL OR i.data_emitere >= :startDate) " +
151 " AND (:endDate IS NULL OR i.data_emitere <= :endDate) " +
152 "GROUP BY adr.strada, adr.localitate " +
153 "ORDER BY nr_incidente DESC",
154 nativeQuery = true)
155 List<Map<String, Object>> getTopStraziIncidente(
156     @Param("startDate") LocalDateTime startDate,
157     @Param("endDate") LocalDateTime endDate
158 );

```

AmendaRepository:

```

80 // Raport "Cazier Fiscal": Istoric amenzi per CNP (JOIN Triplu: Amenzi-Persoane-Politisti)
81 // [Parametri Variabili]: CNP + Interval Data
82 @Query(value = "" + 1usage 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
114 // Agenti Severi: Politistii cu media amenzilor peste media globala a sectiei
115 // [Complexitate]: Subquery in HAVING + AVG
116 // [Parametru Variabil]: Intervalul de date
117 @Query(value = "" + 1 usage new *
118     "SELECT p.nume, p.prenume, p.grad, p.functie, AVG(a.suma) as medie_personala " +
119     "FROM Politisti p " +
120     "JOIN Amenzi a ON p.id_politist = a.id_politist " +
121     "WHERE (:startDate IS NULL OR a.data_emitere >= :startDate) " +
122     "    AND (:endDate IS NULL OR a.data_emitere <= :endDate) " +
123     "GROUP BY p.id_politist, p.nume, p.prenume, p.grad, p.functie " +
124     "HAVING AVG(a.suma) > (" +
125     "    SELECT AVG(a2.suma) " +
126     "    FROM Amenzi a2 " +
127     "    WHERE (:startDate IS NULL OR a2.data_emitere >= :startDate) " +
128     "        AND (:endDate IS NULL OR a2.data_emitere <= :endDate)" +
129     ") " +
130     "ORDER BY medie_personala DESC",
131     nativeQuery = true)
132 List<Map<String, Object>> getAgentiSeveri(
133     @Param("startDate") LocalDateTime startDate,
134     @Param("endDate") LocalDateTime endDate
135 );
```

## PersoanaRepository:

```
100 // Raport "Recidivisti": Persoane cu nr. amenzi peste media populatiei
101 // [Complexitate]: Subquery in HAVING + COUNT/AVG
102 // [Parametri Variabili]: Intervalul de date
103 @Query(value = "" + 1usage new *
104 "SELECT p.nume, p.preume, p.cnp, COUNT(a.id_amenda) as nr_abateri " +
105 "FROM Persoane p " +
106 "JOIN Amenzi a ON p.id_persoana = a.id_persoana " +
107 "WHERE (:startDate IS NULL OR a.data_emitere >= :startDate) " +
108 " AND (:endDate IS NULL OR a.data_emitere <= :endDate) " +
109 "GROUP BY p.id_persoana, p.nume, p.preume, p.cnp " +
110 "HAVING COUNT(a.id_amenda) > (" +
111 "    SELECT CAST(COUNT(*) AS FLOAT) / COUNT(DISTINCT id_persoana) " +
112 "    FROM Amenzi a2 " +
113 "    WHERE (:startDate IS NULL OR a2.data_emitere >= :startDate) " +
114 "    AND (:endDate IS NULL OR a2.data_emitere <= :endDate)" +
115 ") " +
116 "ORDER BY nr_abateri DESC",
117 nativeQuery = true)
118 List<Map<String, Object>> getRecidivisti(
119     @Param("startDate") LocalDateTime startDate,
120     @Param("endDate") LocalDateTime endDate
121 );
```

## IncidentRepository:

```

120 // Raport "Zile Critice": Zile cu criminalitate peste media zilnica globala
121 // [Complexitate]: Subquery in HAVING + AVG + GROUP BY DATE()
122 @Query(value = "" + 1 usage 3 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 // Cerinta: Strazi unde nu s-au inregistrat incidente
162 // Implementare: Selectam strazile a caror ID nu se afla in 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
142 // Search Intelligent (CONCAT + COLLATE)
143 // [Parametru Variabil]: Termenul de cautare
144 @Query(value = "" + 1 usage & vladiv5
145         "SELECT * FROM Politisti WHERE " +
146         "CONCAT(COALESCE(ume, ''), ' ', COALESCE(prenume, ''), ' ', " +
147         "        COALESCE(grad, ''), ' ', COALESCE(funcție, ''), ' ', " +
148         "        COALESCE(telefon_serviciu, '')) " +
149         "COLLATE Latin1_General_100_CI_AI " +
150         "LIKE CONCAT(:termen, '%')",
151         nativeQuery = true)
152 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(ume, ''), ' ', 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 // Validare 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 // Validare 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);
153
```

IncidentRepository:

```

100
101 // Live Search: Cauta in Incidente SI in numele Politistului (LEFT JOIN)
102 // [Parametru Variabil]: Termenul de cautare
103 @Query(value = "" + 1 usage new *
104 "SELECT i.* FROM Incidente i " +
105 "LEFT JOIN Politisti p ON i.id_politist_responsabil = p.id_politist " +
106 "WHERE " +
107 "CONCAT(COALESCE(i.tip_incident, ''), ' ', COALESCE(i.descriere_locatie, ''), ' ', " +
108 "        COALESCE(i.descriere_incident, ''), ' ', COALESCE(p.num, ' '), ' ', " +
109 "        COALESCE(p.prenume, '')) " +
110 "COLLATE Latin1_General_100_CI_AI " +
111 "LIKE CONCAT(:termen, '%')",
112 nativeQuery = true)
113 List<Incident> cautuDupaInceput(@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 2 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.num, ' '), ' ', " +
106 "        COALESCE(p.prenume, ''), ' ', COALESCE(pol.num, ' '), ' ', " +
107 "        COALESCE(pol.prenume, '')) " +
108 "COLLATE Latin1_General_100_CI_AI " +
109 "LIKE CONCAT(:termen, '%')",
110 nativeQuery = true)
111 List<Amenda> cautuDupaInceput(@Param("termen") String termen);
112

```

### AdresaRepository:

```

78 // Search Inteligent: Cauta in toate campurile adresei simultan
79 // Utilizeaza COALESCE pentru a gestiona campurile NULL (ex: bloc/ap)
80 @Query(value = "" + 1 usage 2 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> cautuDupaInceput(@Param("termen") String termen);
89

```