

Proiect SGBD - Gestiune transporturi rutiere

George Mihai Radu - Grupa 244

1 Introducere

Detaliile din acest proiect se referă la proiectarea unui model de date ce furnizează informații despre gestiunea companiilor de transporturi rutiere de mărfuri din România.

Modelul de date va gestiona informații legate de transporturile de mărfuri ale unor companii client de la un depozit la altul. Există firme de pază care se ocupă de paza depozitelor. O firmă de pază alocă fiecărui depozit o echipă de pază.

Transporturile de marfă sunt realizate de o firmă de transport prin intermediul angajaților profesioniști care pot avea mai multe categorii de permise și se pot opri la mai multe popasuri pe traseu. Transportul de marfă începe de la un depozit și se termină la alt depozit, în fiecare depozit accesul camionului este permis după o verificare a echipei de pază.

2 Diagrama entitate-relație E/R

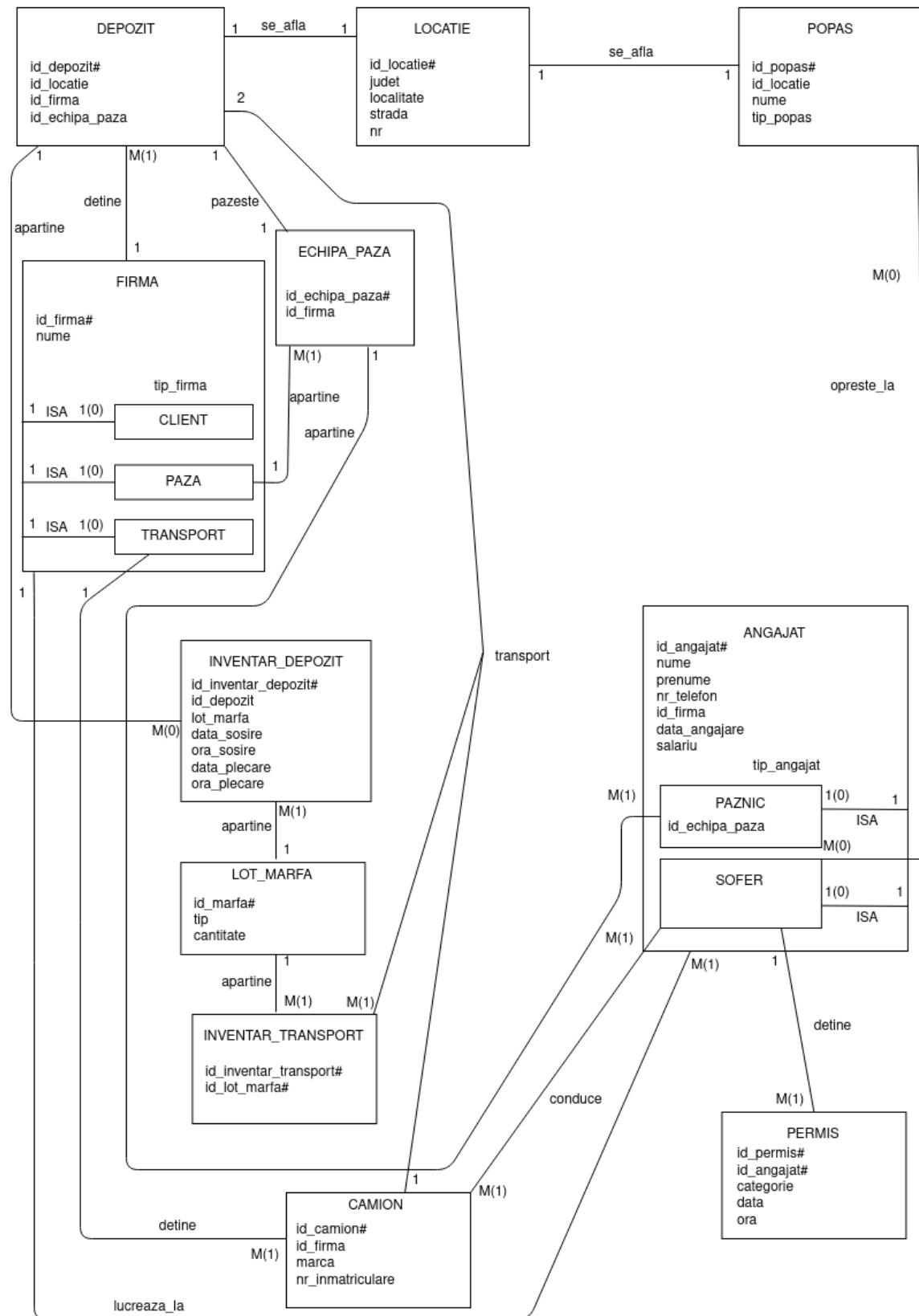


Figura 1

3 Diagrama conceptuală

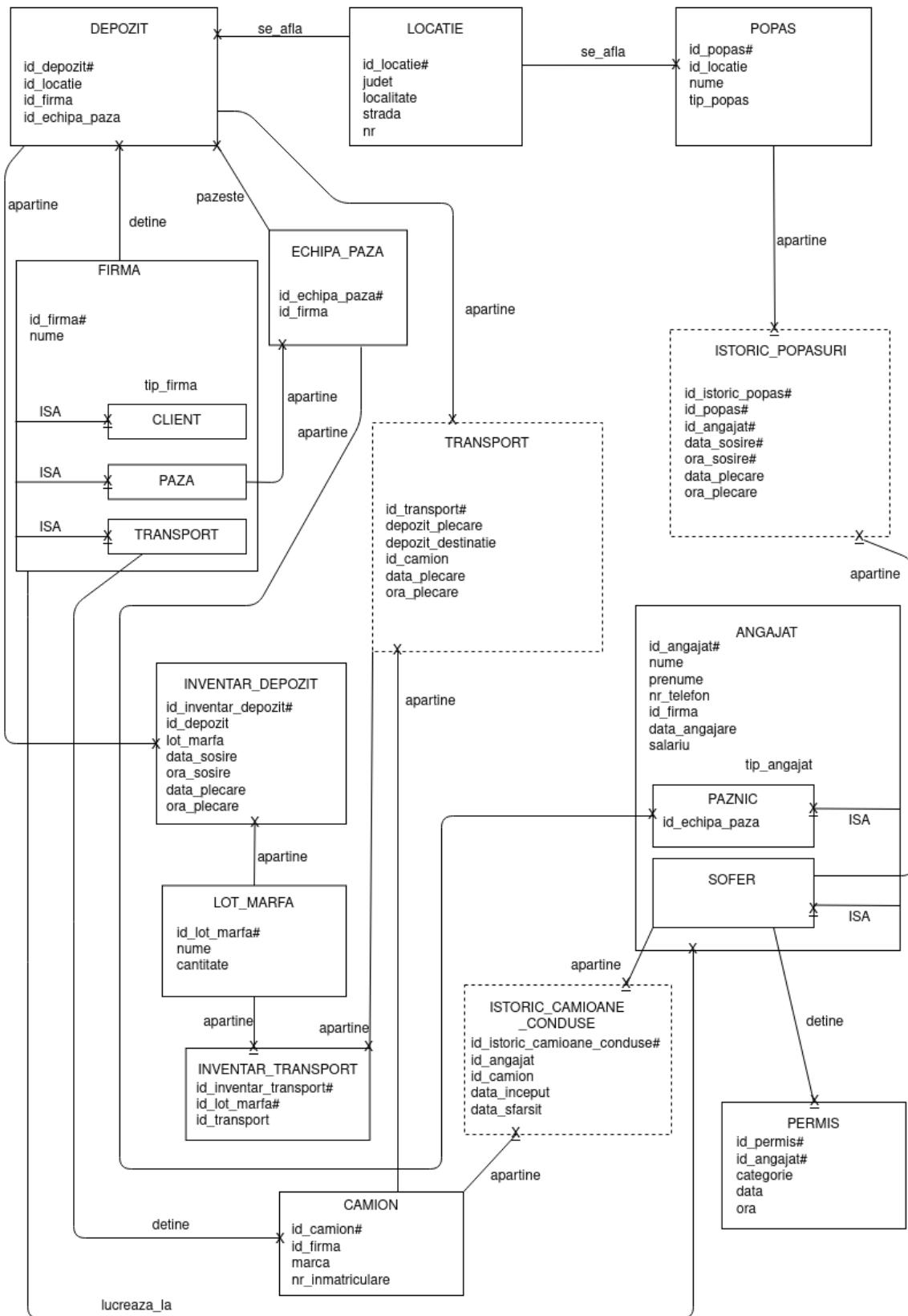


Figura 2

4 Implementarea diagramei conceptuale

```
01 | -- crearea tabelelor
02 |
03 | CREATE TABLE LOCATIE
04 |     (id_locatie number(10) constraint pk_loc primary key,
05 |      judet varchar2(25),
06 |      localitate varchar2(25),
07 |      strada varchar2(25),
08 |      nr number(5)
09 |      );
10 |
11 | CREATE TABLE POPAS
12 |     (id_popas number(10) constraint pk_pop primary key,
13 |      id_locatie number(10) constraint fk_pop_loc references
14 |                               LOCATIE(id_locatie) on delete set null,
15 |      nume varchar2(25),
16 |      tip_popas varchar2(10) constraint tip_pop not null,
17 |      constraint verif_tip_pop check(tip_popas = 'BENZINARIE'
18 |                                       or tip_popas = 'MOTEL' or tip_popas = 'RESTAURANT')
19 |      );
20 |
21 | CREATE TABLE FIRMA
22 |     (id_firma number(12) constraint pk_firma primary key,
23 |      nume varchar2(25) constraint uni_nume_firma unique,
24 |      tip_firma varchar2(10) constraint tip_firma not null,
25 |      constraint verif_tip_firma check(tip_firma = 'CLIENT' or tip_firma = 'PAZA'
26 |                                       or tip_firma = 'TRANSPORT')
27 |      );
28 |
29 | CREATE TABLE ECHIPA_PAZA
30 |     (id_echipa_paza number(5) constraint pk_epaza primary key,
31 |      id_firma number(12) constraint fk_epaza_firma references
32 |                               FIRMA(id_firma) on delete cascade
33 |      );
34 |
35 | CREATE TABLE DEPOZIT
36 |     (id_depozit number(5) constraint pk_dep primary key,
37 |      id_locatie number(5) constraint fk_dep_loc references
38 |                               LOCATIE(id_locatie) on delete set null,
39 |      id_echipa_paza number(5) constraint fk_dep_epaza
40 |      references ECHIPA_PAZA(id_echipa_paza) on delete set null,
41 |      id_firma number(12) constraint fk_dep_firma references FIRMA(id_firma)
42 |      on delete set null
43 |      );
44 |
45 | CREATE TABLE CAMION
46 |     (id_camion number(5) constraint pk_cam primary key,
47 |      marca varchar2(10),
48 |      nr_inmatriculara varchar2(10) constraint uni_nr_ing_cam unique,
49 |      id_firma number(12) constraint fk_cam_firma references
50 |                               FIRMA(id_firma) on delete set null
51 |      );
52 |
53 | CREATE TABLE TRANSPORT
54 |     (id_transport number(10) constraint pk_trans primary key,
55 |      depozit_plecare number(5) constraint fk_trans_dep_p references
56 |                               DEPOZIT(id_depozit) on delete cascade,
57 |      depozit_destinatie number(5) constraint fk_trans_dep_d references
58 |                               DEPOZIT(id_depozit) on delete cascade,
59 |      id_camion number(5) constraint fk_trans_cam references
60 |                               CAMION(id_camion) on delete cascade,
61 |      data_plecare date default sysdate
62 |      );
63 |
64 | CREATE TABLE LOT_MARFA
65 |     (id_lot_marfa number(10) constraint pk_lotm primary key,
66 |      nume varchar2(25) constraint nume_lotm not null,
67 |      cantitate number(8, 2) constraint cant_lotm not null
68 |      );
69 |
70 | CREATE TABLE INVENTAR_DEPOZIT
71 |     (id_inventar_depozit number(10) constraint pk_inv_dep primary key,
```

```

72 |     id_depozit number(5) constraint fk_invdep_dep references
73 |                         DEPOZIT(id_depozit) on delete cascade,
74 |     id_lot_marfa number(10) constraint fk_invdep_lotm references
75 |                         LOT_MARFA(id_lot_marfa) on delete cascade,
76 |     data_sosire date default sysdate,
77 |     data_plecare date
78 | );
79 |
80 | CREATE TABLE INVENTAR_TRANSPORT
81 |     (id_inventar_transport number(10),
82 |      id_lot_marfa number(10) constraint fk_invtran_lotm references
83 |                         LOT_MARFA(id_lot_marfa) on delete cascade,
84 |      id_transport number(10) constraint fk_invtran_tran references
85 |                         TRANSPORT(id_transport) on delete cascade,
86 |      constraint pk_inv_tran primary key(id_inventar_transport, id_lot_marfa)
87 | );
88 |
89 | CREATE TABLE ANGAJAT
90 |     (id_angajat number(10) constraint pk_ang primary key,
91 |      nume varchar(25),
92 |      prenume varchar(25),
93 |      nr_telefon varchar(14),
94 |      data_angajare date default sysdate,
95 |      salariu number(10),
96 |      id_firma number(12),
97 |      tip_angajat char(6) constraint tip_ang_nn not null,
98 |      id_echipa_paza number(5),
99 |      constraint verf_tip check(tip_angajat = 'PAZNIC' or tip_angajat = 'SOFER'),
100 |      constraint fk_ang_epaza foreign key(id_echipa_paza)
101 |          references ECHIPA_PAZA(id_echipa_paza) on delete set null,
102 |      constraint fk_ang_firma foreign key(id_firma)
103 |          references FIRMA(id_firma) on delete set null
104 | );
105 |
106 | CREATE TABLE ISTORIC_CAMIOANE_CONDUSE
107 |     (id_istoric_camioane_conduse number(10) constraint pk_ist_cam primary key,
108 |      id_angajat number(10) constraint fk_icam_ang references
109 |                         ANGAJAT(id_angajat) on delete cascade,
110 |      id_camion number(5) constraint fk_icam_cam references
111 |                         CAMION(id_camion) on delete cascade,
112 |      data_inceput date default sysdate,
113 |      data_sfarsit date
114 | );
115 |
116 | CREATE TABLE PERMIS
117 |     (id_permis number(10),
118 |      id_angajat number(10) constraint fk_perm_ang references
119 |                         ANGAJAT(id_angajat) on delete cascade,
120 |      categorie char(3),
121 |      data date default sysdate,
122 |      constraint pk_perm primary key(id_permis, id_angajat),
123 |      constraint verf_catg check(categorie in ('AM', 'A1', 'A2', 'A', 'B1', 'B',
124 |      'BE', 'C1', 'C1E', 'C', 'CE', 'D1', 'D1E', 'D', 'DE', 'Tr', 'Tb', 'Tv'))
125 | );
126 |
127 | CREATE TABLE ISTORIC_POPASURI
128 |     (id_istoric_popas number(20),
129 |      id_popas number(10) constraint fk_istpop_pop references POPAS(id_popas),
130 |      id_angajat number(10) constraint fk_istpop_ang references ANGAJAT(id_angajat),
131 |      data_sosire date default sysdate,
132 |      data_plecare date default sysdate,
133 |      constraint pk_ist_pop primary key(id_istoric_popas, id_popas, id_angajat,
134 |      ↪ data_sosire)
135 | );
136 |
137 | /* pentru tabelul Locatie se creeaza urmatoarea secventa SEQ_LOC pentru inserarea
138 | * inregistrarilor in tabelul Locatie, se va proceda analog pentru celelalte
139 | * exemple fara a se mai specifica intr-un comentariu */
140 | CREATE SEQUENCE SEQ_LOC
141 | INCREMENT BY 10
142 | START WITH 10
143 | MAXVALUE 1000000000

```

```

144 | NOCYCLE;
145 |
146 | /* se insereaza date in tabelul Locatie folosind secventa anterior creata
147 | * utilizandu-se metoda implicita | se va proceda analog pentru inserarea de
148 | * date in celelalte tabele fara a se mai specifica intr-un comentariu */
149 | INSERT INTO LOCATIE
150 | VALUES(SEQ_LOC.NEXTVAL, 'Prahova', 'Ploiesti', 'Libertatii', 13);
151 |
152 | INSERT INTO LOCATIE
153 | VALUES(SEQ_LOC.NEXTVAL, 'Prahova', 'Ploiesti', 'Lalelelor', 142);
154 |
155 | INSERT INTO LOCATIE
156 | VALUES(SEQ_LOC.NEXTVAL, 'Prahova', 'Ploiesti', 'Primaverii', 412);
157 |
158 | INSERT INTO LOCATIE
159 | VALUES(SEQ_LOC.NEXTVAL, 'Iasi', 'Iasi', 'Unirii', 9991);
160 |
161 | INSERT INTO LOCATIE
162 | VALUES(SEQ_LOC.NEXTVAL, 'Iasi', 'Iasi', 'Mihai Eminescu', 991);
163 |
164 | INSERT INTO LOCATIE
165 | VALUES(SEQ_LOC.NEXTVAL, 'Iasi', 'Iasi', 'Teilor', 131);
166 |
167 | INSERT INTO LOCATIE
168 | VALUES(SEQ_LOC.NEXTVAL, 'Constanta', 'Constanta', 'Teilor', 21);
169 |
170 | INSERT INTO LOCATIE
171 | VALUES(SEQ_LOC.NEXTVAL, 'Constanta', 'Constanta', 'Victoriei', 123);
172 |
173 | INSERT INTO LOCATIE
174 | VALUES(SEQ_LOC.NEXTVAL, 'Constanta', 'Constanta', 'Zorilor', 13);
175 |
176 | INSERT INTO LOCATIE
177 | VALUES(SEQ_LOC.NEXTVAL, 'Tulcea', 'Tulcea', 'Unirii', 213);
178 |
179 | INSERT INTO LOCATIE
180 | VALUES(SEQ_LOC.NEXTVAL, 'Tulcea', 'Tulcea', 'Victoriei', 113);
181 |
182 | INSERT INTO LOCATIE
183 | VALUES(SEQ_LOC.NEXTVAL, 'Ilfov', 'Bucuresti', 'Libertatii', 112);
184 |
185 | INSERT INTO LOCATIE
186 | VALUES(SEQ_LOC.NEXTVAL, 'Ilfov', 'Bucuresti', '1 mai', 1213);
187 |
188 | INSERT INTO LOCATIE
189 | VALUES(SEQ_LOC.NEXTVAL, 'Ilfov', 'Bucuresti', 'Muncii', 1412);
190 |
191 | INSERT INTO LOCATIE
192 | VALUES(SEQ_LOC.NEXTVAL, 'Ilfov', 'Bucuresti', 'Unirii', 212);
193 |
194 | CREATE SEQUENCE SEQ_POP
195 | INCREMENT BY 20
196 | START WITH 100
197 | MAXVALUE 1000000000
198 | NOCYCLE;
199 |
200 | INSERT INTO POPAS
201 | VALUES(SEQ_POP.NEXTVAL, 10, 'Motel Sunday', 'MOTEL');
202 |
203 | INSERT INTO POPAS
204 | VALUES(SEQ_POP.NEXTVAL, 20, 'Restaurant Ceptura', 'RESTAURANT');
205 |
206 | INSERT INTO POPAS
207 | VALUES(SEQ_POP.NEXTVAL, 40, 'Restaurant Grand', 'RESTAURANT');
208 |
209 | INSERT INTO POPAS
210 | VALUES(SEQ_POP.NEXTVAL, 60, 'Restaurant Unirea', 'RESTAURANT');
211 |
212 | INSERT INTO POPAS
213 | VALUES(SEQ_POP.NEXTVAL, 70, 'Conacul dintre vii', 'MOTEL');
214 |
215 | CREATE SEQUENCE SEQ_FIRMA
216 | INCREMENT BY 100

```

```

217 | START WITH 100
218 | MAXVALUE 1000000
219 | NOCYCLE;
220 |
221 | INSERT INTO FIRMA
222 | VALUES(SEQ_FIRMA.NEXTVAL, 'Lidl', 'CLIENT');
223 |
224 | INSERT INTO FIRMA
225 | VALUES(SEQ_FIRMA.NEXTVAL, 'Profi', 'CLIENT');
226 |
227 | INSERT INTO FIRMA
228 | VALUES(SEQ_FIRMA.NEXTVAL, 'ABOUT YOU', 'CLIENT');
229 |
230 | INSERT INTO FIRMA
231 | VALUES(SEQ_FIRMA.NEXTVAL, 'Deichmann', 'CLIENT');
232 |
233 | INSERT INTO FIRMA
234 | VALUES(SEQ_FIRMA.NEXTVAL, 'H and M', 'CLIENT');
235 |
236 | INSERT INTO FIRMA
237 | VALUES(SEQ_FIRMA.NEXTVAL, 'STOP SEAL GUARD', 'PAZA');
238 |
239 | INSERT INTO FIRMA
240 | VALUES(SEQ_FIRMA.NEXTVAL, 'ATMAN PROTECTION', 'PAZA');
241 |
242 | INSERT INTO FIRMA
243 | VALUES(SEQ_FIRMA.NEXTVAL, 'GARDIENII', 'PAZA');
244 |
245 | INSERT INTO FIRMA
246 | VALUES(SEQ_FIRMA.NEXTVAL, 'QUICK PROTECT', 'PAZA');
247 |
248 | INSERT INTO FIRMA
249 | VALUES(SEQ_FIRMA.NEXTVAL, 'JOHNNY VIP SECURITY', 'PAZA');
250 |
251 | INSERT INTO FIRMA
252 | VALUES(SEQ_FIRMA.NEXTVAL, 'Road Logistics', 'TRANSPORT');
253 |
254 | INSERT INTO FIRMA
255 | VALUES(SEQ_FIRMA.NEXTVAL, 'Transibo', 'TRANSPORT');
256 |
257 | INSERT INTO FIRMA
258 | VALUES(SEQ_FIRMA.NEXTVAL, 'EasyCargo', 'TRANSPORT');
259 |
260 | INSERT INTO FIRMA
261 | VALUES(SEQ_FIRMA.NEXTVAL, 'Lextom Trans Asd', 'TRANSPORT');
262 |
263 | INSERT INTO FIRMA
264 | VALUES(SEQ_FIRMA.NEXTVAL, 'FedEx', 'TRANSPORT');
265 |
266 | INSERT INTO ECHIPA_PAZA
267 | VALUES(1, 600);
268 |
269 | INSERT INTO ECHIPA_PAZA
270 | VALUES(2, 600);
271 |
272 | INSERT INTO ECHIPA_PAZA
273 | VALUES(3, 800);
274 |
275 | INSERT INTO ECHIPA_PAZA
276 | VALUES(4, 700);
277 |
278 | INSERT INTO ECHIPA_PAZA
279 | VALUES(5, 900);
280 |
281 | INSERT INTO ECHIPA_PAZA
282 | VALUES(6, 1000);
283 |
284 | INSERT INTO ECHIPA_PAZA
285 | VALUES(7, 1000);
286 |
287 | INSERT INTO ECHIPA_PAZA
288 | VALUES(8, 600);
289 |

```

```

290 | INSERT INTO ECHIPA_PAZA
291 | VALUES(9, 700);
292 |
293 | INSERT INTO ECHIPA_PAZA
294 | VALUES(10, 600);
295 |
296 | CREATE SEQUENCE SEQ_DEP
297 | INCREMENT BY 500
298 | START WITH 500
299 | MAXVALUE 50000
300 | NOCYCLE;
301 |
302 | INSERT INTO DEPOZIT
303 | VALUES(SEQ_DEP.NEXTVAL, 30, 2, 100);
304 | INSERT INTO DEPOZIT
305 | VALUES(SEQ_DEP.NEXTVAL, 50, 3, 200);
306 | INSERT INTO DEPOZIT
307 | VALUES(SEQ_DEP.NEXTVAL, 80, 1, 300);
308 | INSERT INTO DEPOZIT
309 | VALUES(SEQ_DEP.NEXTVAL, 90, 5, 400);
310 | INSERT INTO DEPOZIT
311 | VALUES(SEQ_DEP.NEXTVAL, 110, 6, 500);
312 | INSERT INTO DEPOZIT
313 | VALUES(SEQ_DEP.NEXTVAL, 100, 4, 1100);
314 | INSERT INTO DEPOZIT
315 | VALUES(SEQ_DEP.NEXTVAL, 120, 10, 1200);
316 | INSERT INTO DEPOZIT
317 | VALUES(SEQ_DEP.NEXTVAL, 140, 8, 1300);
318 | INSERT INTO DEPOZIT
319 | VALUES(SEQ_DEP.NEXTVAL, 150, 7, 1400);
320 | INSERT INTO DEPOZIT
321 | VALUES(SEQ_DEP.NEXTVAL, 130, 9, 1500);
322 |
323 | CREATE SEQUENCE SEQ_CAM
324 | INCREMENT BY 10
325 | START WITH 15
326 | MAXVALUE 10000
327 | NOCYCLE;
328 |
329 | INSERT INTO CAMION
330 | VALUES(SEQ_CAM.NEXTVAL, 'VOLVO', 'CT12FSD', 1100);
331 |
332 | INSERT INTO CAMION
333 | VALUES(SEQ_CAM.NEXTVAL, 'VOLVO', 'CT90MMM', 1100);
334 |
335 | INSERT INTO CAMION
336 | VALUES(SEQ_CAM.NEXTVAL, 'SCANIA', 'PH13SCN', 1200);
337 |
338 | INSERT INTO CAMION
339 | VALUES(SEQ_CAM.NEXTVAL, 'DAF', 'B55ASD', 1300);
340 |
341 | INSERT INTO CAMION
342 | VALUES(SEQ_CAM.NEXTVAL, 'IVECO', 'MM13IVC', 1400);
343 |
344 | INSERT INTO CAMION
345 | VALUES(SEQ_CAM.NEXTVAL, 'MAN', 'TM20QWE', 1100);
346 |
347 | INSERT INTO CAMION
348 | VALUES(SEQ_CAM.NEXTVAL, 'SCANIA', 'IS32ISA', 1100);
349 |
350 | INSERT INTO CAMION
351 | VALUES(SEQ_CAM.NEXTVAL, 'MAN', 'ISO1MAN', 1200);
352 |
353 | INSERT INTO CAMION
354 | VALUES(SEQ_CAM.NEXTVAL, 'DAF', 'TM11IOP', 1300);
355 |
356 | INSERT INTO CAMION
357 | VALUES(SEQ_CAM.NEXTVAL, 'VOLVO', 'MM57RXF', 1500);
358 |
359 | INSERT INTO CAMION
360 | VALUES(SEQ_CAM.NEXTVAL, 'SCANIA', 'B29SCN', 1500);
361 |
362 | CREATE SEQUENCE SEQ_TRAN

```

```

363 | INCREMENT BY 10
364 | START WITH 10000
365 | MAXVALUE 2000000000
366 | NOCYCLE;
367 |
368 | INSERT INTO TRANSPORT
369 | VALUES(SEQ_TRAN.NEXTVAL, 3000, 500, 15, to_date('06-05-2021 08:00', 'dd-mm-yyyy, hh24:
370 |   ↪ mi'));
370 |
371 | INSERT INTO TRANSPORT
372 | VALUES(SEQ_TRAN.NEXTVAL, 3000, 500, 25, to_date('08-05-2021 06:00', 'dd-mm-yyyy, hh24:
373 |   ↪ mi'));
373 |
374 | INSERT INTO TRANSPORT
375 | VALUES(SEQ_TRAN.NEXTVAL, 3500, 1000, 35, to_date('06-05-2021 08:00', 'dd-mm-yyyy, hh24:
376 |   ↪ mi'));
376 |
377 | INSERT INTO TRANSPORT
378 | VALUES(SEQ_TRAN.NEXTVAL, 4000, 1000, 45, to_date('06-05-2021 06:00', 'dd-mm-yyyy, hh24:
379 |   ↪ mi'));
379 |
380 | INSERT INTO TRANSPORT
381 | VALUES(SEQ_TRAN.NEXTVAL, 4500, 1000, 55, to_date('06-05-2021 06:00', 'dd-mm-yyyy, hh24:
382 |   ↪ mi'));
382 |
383 | INSERT INTO TRANSPORT
384 | VALUES(SEQ_TRAN.NEXTVAL, 5000, 1500, 115, to_date('07-05-2021 08:00', 'dd-mm-yyyy, hh24
385 |   ↪ :mi'));
385 |
386 | INSERT INTO TRANSPORT
387 | VALUES(SEQ_TRAN.NEXTVAL, 5000, 1500, 105, to_date('05-05-2021 06:00', 'dd-mm-yyyy, hh24
388 |   ↪ :mi'));
388 |
389 | INSERT INTO TRANSPORT
390 | VALUES(SEQ_TRAN.NEXTVAL, 3500, 2000, 85, to_date('06-05-2021 06:00', 'dd-mm-yyyy, hh24:
391 |   ↪ mi'));
391 |
392 | INSERT INTO TRANSPORT
393 | VALUES(SEQ_TRAN.NEXTVAL, 3500, 2000, 85, to_date('07-05-2021 06:00', 'dd-mm-yyyy, hh24:
394 |   ↪ mi'));
394 |
395 | INSERT INTO TRANSPORT
396 | VALUES(SEQ_TRAN.NEXTVAL, 3000, 2500, 25, to_date('10-05-2021 07:00', 'dd-mm-yyyy, hh24:
397 |   ↪ mi'));
397 |
398 | INSERT INTO TRANSPORT
399 | VALUES(SEQ_TRAN.NEXTVAL, 3000, 2500, 15, to_date('11-05-2021 05:00', 'dd-mm-yyyy, hh24:
400 |   ↪ mi'));
400 |
401 | CREATE SEQUENCE SEQ_LOTM
402 | INCREMENT BY 5
403 | START WITH 100
404 | MAXVALUE 5000000000
405 | NOCYCLE;
406 |
407 | INSERT INTO LOT_MARFA
408 | VALUES(SEQ_LOTM.NEXTVAL, 'carne miel', 100.00);
409 |
410 | INSERT INTO LOT_MARFA
411 | VALUES(SEQ_LOTM.NEXTVAL, 'carne porc', 300.00);
412 |
413 | INSERT INTO LOT_MARFA
414 | VALUES(SEQ_LOTM.NEXTVAL, 'carne pui', 250.00);
415 |
416 | INSERT INTO LOT_MARFA
417 | VALUES(SEQ_LOTM.NEXTVAL, 'carne pui', 200.00);
418 |
419 | INSERT INTO LOT_MARFA
420 | VALUES(SEQ_LOTM.NEXTVAL, 'rosii', 1000.00);
421 |
422 | INSERT INTO LOT_MARFA
423 | VALUES(SEQ_LOTM.NEXTVAL, 'rosii', 1500.00);
424 |

```

```

425 | INSERT INTO LOT_MARFA
426 | VALUES(SEQ_LOTM.NEXTVAL, 'ardei', 500.00);
427 |
428 | INSERT INTO LOT_MARFA
429 | VALUES(SEQ_LOTM.NEXTVAL, 'lapte', 100.00);
430 |
431 | INSERT INTO LOT_MARFA
432 | VALUES(SEQ_LOTM.NEXTVAL, 'lapte', 100.00);
433 |
434 | INSERT INTO LOT_MARFA
435 | VALUES(SEQ_LOTM.NEXTVAL, 'lapte', 120.00);
436 |
437 | INSERT INTO LOT_MARFA
438 | VALUES(SEQ_LOTM.NEXTVAL, 'tricouri', 100.00);
439 |
440 | INSERT INTO LOT_MARFA
441 | VALUES(SEQ_LOTM.NEXTVAL, 'tricouri', 90.00);
442 |
443 | INSERT INTO LOT_MARFA
444 | VALUES(SEQ_LOTM.NEXTVAL, 'tricouri', 10.00);
445 |
446 | INSERT INTO LOT_MARFA
447 | VALUES(SEQ_LOTM.NEXTVAL, 'pantofi', 20.00);
448 |
449 | INSERT INTO LOT_MARFA
450 | VALUES(SEQ_LOTM.NEXTVAL, 'pantofi', 30.00);
451 |
452 | INSERT INTO LOT_MARFA
453 | VALUES(SEQ_LOTM.NEXTVAL, 'hanorace', 50.00);
454 |
455 | INSERT INTO LOT_MARFA
456 | VALUES(SEQ_LOTM.NEXTVAL, 'hanorace', 35.00);
457 |
458 | INSERT INTO LOT_MARFA
459 | VALUES(SEQ_LOTM.NEXTVAL, 'rochii', 50.00);
460 |
461 | INSERT INTO LOT_MARFA
462 | VALUES(SEQ_LOTM.NEXTVAL, 'fuste', 35.00);
463 |
464 | INSERT INTO LOT_MARFA
465 | VALUES(SEQ_LOTM.NEXTVAL, 'blugi', 35.00);
466 |
467 | CREATE SEQUENCE SEQ_INV_DEP
468 | INCREMENT BY 50
469 | START WITH 100
470 | MAXVALUE 500000000
471 | NOCYCLE;
472 |
473 | --10040
474 | INSERT INTO INVENTAR_DEPOZIT
475 | VALUES(SEQ_INV_DEP.NEXTVAL, 3000, 100, to_date('02-04-2021 12:00', 'dd-mm-yyyy hh24:mi'
476 |   ↪ ), null);
476 | --10040
477 | INSERT INTO INVENTAR_DEPOZIT
478 | VALUES(SEQ_INV_DEP.NEXTVAL, 3000, 105, to_date('02-04-2021 10:00', 'dd-mm-yyyy hh24:mi'
479 |   ↪ ), null);
479 | --10050
480 | INSERT INTO INVENTAR_DEPOZIT
481 | VALUES(SEQ_INV_DEP.NEXTVAL, 3000, 110, to_date('01-04-2021 11:00', 'dd-mm-yyyy hh24:mi'
482 |   ↪ ), null);
482 |
483 | INSERT INTO INVENTAR_DEPOZIT
484 | VALUES(SEQ_INV_DEP.NEXTVAL, 3000, 115, to_date('25-04-2021 12:00', 'dd-mm-yyyy hh24:mi'
485 |   ↪ ), null);
485 |
486 | --10060
487 | INSERT INTO INVENTAR_DEPOZIT
488 | VALUES(SEQ_INV_DEP.NEXTVAL, 3500, 120, to_date('24-04-2021 12:00', 'dd-mm-yyyy hh24:mi'
489 |   ↪ ), null);
489 | INSERT INTO INVENTAR_DEPOZIT
490 | VALUES(SEQ_INV_DEP.NEXTVAL, 3500, 130, to_date('07-04-2021 12:00', 'dd-mm-yyyy hh24:mi'
491 |   ↪ ), null);

```

```

492 | --10070
493 | INSERT INTO INVENTAR_DEPOZIT
494 | VALUES(SEQ_INV_DEP.NEXTVAL, 4000, 140, to_date('12-04-2021 12:00', 'dd-mm-yyyy hh24:mi'
495 |   ), null);
496 | --10080
497 | INSERT INTO INVENTAR_DEPOZIT
498 | VALUES(SEQ_INV_DEP.NEXTVAL, 4500, 125, to_date('21-04-2021 12:00', 'dd-mm-yyyy hh24:mi'
499 |   ), null);
500 | INSERT INTO INVENTAR_DEPOZIT
501 | VALUES(SEQ_INV_DEP.NEXTVAL, 4500, 145, to_date('13-04-2021 12:00', 'dd-mm-yyyy hh24:mi'
502 |   ), null);
503 | --10090
504 | INSERT INTO INVENTAR_DEPOZIT
505 | VALUES(SEQ_INV_DEP.NEXTVAL, 5000, 195, to_date('11-04-2021 12:00', 'dd-mm-yyyy hh24:mi'
506 |   ), null);
507 | --10100
508 | INSERT INTO INVENTAR_DEPOZIT
509 | VALUES(SEQ_INV_DEP.NEXTVAL, 5000, 150, to_date('14-04-2021 12:00', 'dd-mm-yyyy hh24:mi'
510 |   ), null);
511 | INSERT INTO INVENTAR_DEPOZIT
512 | VALUES(SEQ_INV_DEP.NEXTVAL, 5000, 175, to_date('02-04-2021 12:00', 'dd-mm-yyyy hh24:mi'
513 |   ), null);
514 | --10110
515 | INSERT INTO INVENTAR_DEPOZIT
516 | VALUES(SEQ_INV_DEP.NEXTVAL, 3500, 165, to_date('03-04-2021 12:00', 'dd-mm-yyyy hh24:mi'
517 |   ), null);
518 | --10120
519 | INSERT INTO INVENTAR_DEPOZIT
520 | VALUES(SEQ_INV_DEP.NEXTVAL, 3500, 170, to_date('02-04-2021 12:00', 'dd-mm-yyyy hh24:mi'
521 |   ), null);
522 | INSERT INTO INVENTAR_DEPOZIT
523 | VALUES(SEQ_INV_DEP.NEXTVAL, 3000, 190, to_date('06-04-2021 12:00', 'dd-mm-yyyy hh24:mi'
524 |   ), null);
525 | --10130
526 | INSERT INTO INVENTAR_DEPOZIT
527 | VALUES(SEQ_INV_DEP.NEXTVAL, 3000, 185, to_date('01-04-2021 12:00', 'dd-mm-yyyy hh24:mi'
528 |   ), null);
529 | --10140
530 | INSERT INTO INVENTAR_DEPOZIT
531 | VALUES(SEQ_INV_DEP.NEXTVAL, 5000, 160, to_date('11-04-2021 12:00', 'dd-mm-yyyy hh24:mi'
532 |   ), null);
533 | INSERT INTO INVENTAR_DEPOZIT
534 | VALUES(SEQ_INV_DEP.NEXTVAL, 5000, 155, to_date('11-04-2021 12:00', 'dd-mm-yyyy hh24:mi'
535 |   ), null);
536 | INSERT INTO INVENTAR_DEPOZIT
537 | VALUES(SEQ_INV_DEP.NEXTVAL, 5000, 135, to_date('11-04-2021 12:00', 'dd-mm-yyyy hh24:mi'
538 |   ), null);
539 | CREATE SEQUENCE SEQ_INV_TRAN
540 | INCREMENT BY 55
541 | START WITH 100
542 | MAXVALUE 500000000
543 | NOCYCLE;
544 | INSERT INTO INVENTAR_TRANSPORT
545 | VALUES(SEQ_INV_TRAN.NEXTVAL, 100, 10000);
546 | INSERT INTO INVENTAR_TRANSPORT
547 | VALUES(SEQ_INV_TRAN.NEXTVAL, 105, 10000);
548 | INSERT INTO INVENTAR_TRANSPORT
549 | VALUES(SEQ_INV_TRAN.NEXTVAL, 110, 10010);

```

```

551 |
552 | INSERT INTO INVENTAR_TRANSPORT
553 | VALUES(SEQ_INV_TRAN.NEXTVAL, 120, 10020);
554 | INSERT INTO INVENTAR_TRANSPORT
555 | VALUES(SEQ_INV_TRAN.NEXTVAL, 130, 10020);
556 |
557 | INSERT INTO INVENTAR_TRANSPORT
558 | VALUES(SEQ_INV_TRAN.NEXTVAL, 140, 10030);
559 |
560 | INSERT INTO INVENTAR_TRANSPORT
561 | VALUES(SEQ_INV_TRAN.NEXTVAL, 125, 10040);
562 | INSERT INTO INVENTAR_TRANSPORT
563 | VALUES(SEQ_INV_TRAN.NEXTVAL, 145, 10040);
564 |
565 | INSERT INTO INVENTAR_TRANSPORT
566 | VALUES(SEQ_INV_TRAN.NEXTVAL, 195, 10050);
567 |
568 | INSERT INTO INVENTAR_TRANSPORT
569 | VALUES(SEQ_INV_TRAN.NEXTVAL, 150, 10060);
570 | INSERT INTO INVENTAR_TRANSPORT
571 | VALUES(SEQ_INV_TRAN.NEXTVAL, 175, 10060);
572 |
573 | INSERT INTO INVENTAR_TRANSPORT
574 | VALUES(SEQ_INV_TRAN.NEXTVAL, 165, 10070);
575 |
576 | INSERT INTO INVENTAR_TRANSPORT
577 | VALUES(SEQ_INV_TRAN.NEXTVAL, 170, 10080);
578 |
579 | INSERT INTO INVENTAR_TRANSPORT
580 | VALUES(SEQ_INV_TRAN.NEXTVAL, 190, 10090);
581 | INSERT INTO INVENTAR_TRANSPORT
582 | VALUES(SEQ_INV_TRAN.NEXTVAL, 185, 10090);
583 |
584 | INSERT INTO INVENTAR_TRANSPORT
585 | VALUES(SEQ_INV_TRAN.NEXTVAL, 180, 10100);
586 |
587 | CREATE SEQUENCE SEQ_ANG
588 | INCREMENT BY 20
589 | START WITH 20000
590 | MAXVALUE 2000000000
591 | NOCYCLE;
592 |
593 | INSERT INTO ANGAJAT
594 | VALUES(SEQ_ANG.NEXTVAL, 'Ion', 'Vasile', '0722123456',
595 |         to_date('11-04-2020 12:00', 'dd-mm-yyyy hh24:mi'), 5000, 1100, 'SOFER', null);
596 |
597 | INSERT INTO ANGAJAT
598 | VALUES(SEQ_ANG.NEXTVAL, 'Ovidiu', 'Ion', '0727124356',
599 |         to_date('12-05-2020 12:00', 'dd-mm-yyyy hh24:mi'), 4000, 1100, 'SOFER', null);
600 |
601 | INSERT INTO ANGAJAT
602 | VALUES(SEQ_ANG.NEXTVAL, 'Popescu', 'Andrei', '0723323465',
603 |         to_date('11-03-2020 12:00', 'dd-mm-yyyy hh24:mi'), 6000, 1200, 'SOFER', null);
604 |
605 | INSERT INTO ANGAJAT
606 | VALUES(SEQ_ANG.NEXTVAL, 'Marian', 'Gheorghe', '0721213456',
607 |         to_date('11-11-2020 12:00', 'dd-mm-yyyy hh24:mi'), 3000, 1200, 'SOFER', null);
608 |
609 | INSERT INTO ANGAJAT
610 | VALUES(SEQ_ANG.NEXTVAL, 'Manea', 'George', '0721123959',
611 |         to_date('03-03-2020 12:00', 'dd-mm-yyyy hh24:mi'), 4000, 1300, 'SOFER', null);
612 |
613 | INSERT INTO ANGAJAT
614 | VALUES(SEQ_ANG.NEXTVAL, 'Constantin', 'Gheorghe', '0735123456',
615 |         to_date('09-04-2020 12:00', 'dd-mm-yyyy hh24:mi'), 4500, 1400, 'SOFER', null);
616 |
617 | INSERT INTO ANGAJAT
618 | VALUES(SEQ_ANG.NEXTVAL, 'Ion', 'Catalin', '0721126566',
619 |         to_date('02-07-2020 12:00', 'dd-mm-yyyy hh24:mi'), 3400, 1500, 'SOFER', null);
620 |
621 | INSERT INTO ANGAJAT
622 | VALUES(SEQ_ANG.NEXTVAL, 'Marius', 'Catalin', '072119356',
623 |         to_date('02-07-2020 12:00', 'dd-mm-yyyy hh24:mi'), 3300, 600, 'PAZNIC', 1);

```

```

624 | INSERT INTO ANGAJAT
625 | VALUES(SEQ_ANG.NEXTVAL, 'Grant', 'Ion', '0723123456',
626 |        to_date('02-07-2020 12:00', 'dd-mm-yyyy hh24:mi'), 3100, 600, 'PAZNIC', 1);
627 |
628 | INSERT INTO ANGAJAT
629 | VALUES(SEQ_ANG.NEXTVAL, 'Ivan', 'George', '0726100456',
630 |        to_date('02-07-2020 12:00', 'dd-mm-yyyy hh24:mi'), 3250, 600, 'PAZNIC', 2);
631 |
632 | INSERT INTO ANGAJAT
633 | VALUES(SEQ_ANG.NEXTVAL, 'Neagu', 'Ion', '0728123456',
634 |        to_date('02-07-2020 12:00', 'dd-mm-yyyy hh24:mi'), 3500, 800, 'PAZNIC', 3);
635 |
636 | INSERT INTO ANGAJAT
637 | VALUES(SEQ_ANG.NEXTVAL, 'Popescu', 'Cristian', '0724523456',
638 |        to_date('02-07-2020 12:00', 'dd-mm-yyyy hh24:mi'), 3600, 700, 'PAZNIC', 4);
639 |
640 | INSERT INTO ANGAJAT
641 | VALUES(SEQ_ANG.NEXTVAL, 'Marius', 'Mihai', '0721223456',
642 |        to_date('02-07-2020 12:00', 'dd-mm-yyyy hh24:mi'), 3450, 900, 'PAZNIC', 5);
643 |
644 | INSERT INTO ANGAJAT
645 | VALUES(SEQ_ANG.NEXTVAL, 'Radu', 'Gheorghe', '0724323456',
646 |        to_date('02-07-2020 12:00', 'dd-mm-yyyy hh24:mi'), 3375, 1000, 'PAZNIC', 6);
647 |
648 | INSERT INTO ANGAJAT
649 | VALUES(SEQ_ANG.NEXTVAL, 'Marin', 'Catalin', '0725623456',
650 |        to_date('02-07-2020 12:00', 'dd-mm-yyyy hh24:mi'), 3150, 1000, 'PAZNIC', 7);
651 |
652 | INSERT INTO ANGAJAT
653 | VALUES(SEQ_ANG.NEXTVAL, 'Andrei', 'David', '0722123456',
654 |        to_date('02-07-2020 12:00', 'dd-mm-yyyy hh24:mi'), 3250, 600, 'PAZNIC', 8);
655 |
656 | INSERT INTO ANGAJAT
657 | VALUES(SEQ_ANG.NEXTVAL, 'Popescu', 'Stefan', '0721145456',
658 |        to_date('02-07-2020 12:00', 'dd-mm-yyyy hh24:mi'), 3400, 700, 'PAZNIC', 9);
659 |
660 | INSERT INTO ANGAJAT
661 | VALUES(SEQ_ANG.NEXTVAL, 'Ion', 'Catalin', '0721765456',
662 |        to_date('02-07-2020 12:00', 'dd-mm-yyyy hh24:mi'), 3320, 600, 'PAZNIC', 10);
663 |
664 | CREATE SEQUENCE SEQ_IST_CAM
665 | INCREMENT BY 5
666 | START WITH 5
667 | MAXVALUE 2000000000
668 | NOCYCLE;
669 |
670 | INSERT INTO ISTORIC_CAMIOANE_CONDUSE
671 | VALUES(SEQ_IST_CAM.NEXTVAL, 20000, 65, to_date('11-04-2020 12:00', 'dd-mm-yyyy hh24:mi'
672 |   ↪ ),
673 |   to_date('13-07-2020 12:00', 'dd-mm-yyyy hh24:mi'));
674 |
675 | INSERT INTO ISTORIC_CAMIOANE_CONDUSE
676 | VALUES(SEQ_IST_CAM.NEXTVAL, 20000, 75, to_date('13-07-2020 12:00', 'dd-mm-yyyy hh24:mi'
677 |   ↪ ),
678 |   to_date('21-11-2020 12:00', 'dd-mm-yyyy hh24:mi'));
679 |
680 | INSERT INTO ISTORIC_CAMIOANE_CONDUSE
681 | VALUES(SEQ_IST_CAM.NEXTVAL, 20000, 15, to_date('21-11-2020 12:00', 'dd-mm-yyyy hh24:mi'
682 |   ↪ ), null);
683 |
684 | INSERT INTO ISTORIC_CAMIOANE_CONDUSE
685 | VALUES(SEQ_IST_CAM.NEXTVAL, 20020, 25, to_date('12-05-2020 12:00', 'dd-mm-yyyy hh24:mi'
686 |   ↪ ), null);
687 |
688 | INSERT INTO ISTORIC_CAMIOANE_CONDUSE
689 | VALUES(SEQ_IST_CAM.NEXTVAL, 20040, 85, to_date('11-03-2020 12:00', 'dd-mm-yyyy hh24:mi'
690 |   ↪ ), null);

```

```

691 | VALUES(SEQ_IST_CAM.NEXTVAL, 20080, 45, to_date('03-03-2020 12:00', 'dd-mm-yyyy hh24:mi'
692 |   ↪ ), null);
693 | INSERT INTO ISTORIC_CAMIOANE_CONDUSE
694 | VALUES(SEQ_IST_CAM.NEXTVAL, 20100, 55, to_date('09-04-2020 12:00', 'dd-mm-yyyy hh24:mi'
695 |   ↪ ), null);
696 | INSERT INTO ISTORIC_CAMIOANE_CONDUSE
697 | VALUES(SEQ_IST_CAM.NEXTVAL, 20120, 105, to_date('02-07-2020 12:00', 'dd-mm-yyyy hh24:mi'
698 |   ↪ '),
699 |     to_date('06-06-2020 12:00', 'dd-mm-yyyy hh24:mi'));
700 | INSERT INTO ISTORIC_CAMIOANE_CONDUSE
701 | VALUES(SEQ_IST_CAM.NEXTVAL, 20120, 115, to_date('06-06-2020 12:00', 'dd-mm-yyyy hh24:mi'
702 |   ↪ '), null);
703 | CREATE SEQUENCE SEQ_PERM
704 | INCREMENT BY 2
705 | START WITH 1
706 | MAXVALUE 1000000000
707 | NOCYCLE;
708 |
709 | INSERT INTO PERMIS
710 | VALUES(SEQ_PERM.NEXTVAL, 20000, 'C', to_date('06-06-2017 12:00', 'dd-mm-yyyy hh24:mi'));
711 |
712 | INSERT INTO PERMIS
713 | VALUES(SEQ_PERM.NEXTVAL, 20000, 'A', to_date('06-06-2013 12:00', 'dd-mm-yyyy hh24:mi'));
714 |
715 | INSERT INTO PERMIS
716 | VALUES(SEQ_PERM.NEXTVAL, 20000, 'B', to_date('06-06-2012 12:00', 'dd-mm-yyyy hh24:mi'));
717 |
718 | INSERT INTO PERMIS
719 | VALUES(SEQ_PERM.NEXTVAL, 20020, 'C', to_date('06-06-2018 12:00', 'dd-mm-yyyy hh24:mi'));
720 |
721 | INSERT INTO PERMIS
722 | VALUES(SEQ_PERM.NEXTVAL, 20040, 'C', to_date('06-05-2016 12:00', 'dd-mm-yyyy hh24:mi'));
723 |
724 | INSERT INTO PERMIS
725 | VALUES(SEQ_PERM.NEXTVAL, 20060, 'C', to_date('03-11-2014 12:00', 'dd-mm-yyyy hh24:mi'));
726 |
727 | INSERT INTO PERMIS
728 | VALUES(SEQ_PERM.NEXTVAL, 20080, 'C', to_date('03-03-2002 12:00', 'dd-mm-yyyy hh24:mi'));
729 |
730 | INSERT INTO PERMIS
731 | VALUES(SEQ_PERM.NEXTVAL, 20100, 'B', to_date('06-11-2010 12:00', 'dd-mm-yyyy hh24:mi'));
732 |
733 | INSERT INTO PERMIS
734 | VALUES(SEQ_PERM.NEXTVAL, 20100, 'C', to_date('06-11-2012 12:00', 'dd-mm-yyyy hh24:mi'));
735 |
736 | INSERT INTO PERMIS
737 | VALUES(SEQ_PERM.NEXTVAL, 20120, 'C', to_date('12-06-2014 12:00', 'dd-mm-yyyy hh24:mi'));
738 |
739 | CREATE SEQUENCE SEQ_IST_POP
740 | INCREMENT BY 5
741 | START WITH 50
742 | MAXVALUE 100000000000000000000000
743 | NOCYCLE;
744 |
745 | INSERT INTO ISTORIC_POPASURI
746 | VALUES(SEQ_IST_POP.NEXTVAL, 180, 20000, to_date('06-05-2021 10:00', 'dd-mm-yyyy hh24:mi'
747 |   ↪ '), to_date('06-05-2021 12:00', 'dd-mm-yyyy hh24:mi'));
748 | INSERT INTO ISTORIC_POPASURI
749 | VALUES(SEQ_IST_POP.NEXTVAL, 120, 20000, to_date('06-05-2021 21:00', 'dd-mm-yyyy hh24:mi'
750 |   ↪ '), to_date('06-05-2021 21:30', 'dd-mm-yyyy hh24:mi'));
751 | INSERT INTO ISTORIC_POPASURI
752 | VALUES(SEQ_IST_POP.NEXTVAL, 180, 20020, to_date('08-05-2021 10:00', 'dd-mm-yyyy hh24:mi'
753 |   ↪ '), to_date('08-05-2021 12:00', 'dd-mm-yyyy hh24:mi'));
754 | INSERT INTO ISTORIC_POPASURI
755 | VALUES(SEQ_IST_POP.NEXTVAL, 120, 20020, to_date('08-05-2021 21:00', 'dd-mm-yyyy hh24:mi'
    ↪ '), to_date('08-05-2021 21:30', 'dd-mm-yyyy hh24:mi'));

```

```

756 |
757 | INSERT INTO ISTORIC_POPASURI
758 | VALUES(SEQ_IST_POP.NEXTVAL, 100, 20060, to_date('06-05-2021 10:00', 'dd-mm-yyyy hh24:mi
    ↵ '), to_date('06-05-2021 10:20', 'dd-mm-yyyy hh24:mi'));
759 |
760 | INSERT INTO ISTORIC_POPASURI
761 | VALUES(SEQ_IST_POP.NEXTVAL, 140, 20060, to_date('06-05-2021 19:00', 'dd-mm-yyyy hh24:mi
    ↵ '), to_date('06-05-2021 19:30', 'dd-mm-yyyy hh24:mi'));
762 |
763 | INSERT INTO ISTORIC_POPASURI
764 | VALUES(SEQ_IST_POP.NEXTVAL, 140, 20080, to_date('06-05-2021 17:00', 'dd-mm-yyyy hh24:mi
    ↵ '), to_date('06-05-2021 17:30', 'dd-mm-yyyy hh24:mi'));
765 |
766 | INSERT INTO ISTORIC_POPASURI
767 | VALUES(SEQ_IST_POP.NEXTVAL, 140, 20100, to_date('06-05-2021 17:00', 'dd-mm-yyyy hh24:mi
    ↵ '), to_date('06-05-2021 17:30', 'dd-mm-yyyy hh24:mi'));
768 |
769 | INSERT INTO ISTORIC_POPASURI
770 | VALUES(SEQ_IST_POP.NEXTVAL, 180, 20120, to_date('07-05-2021 17:00', 'dd-mm-yyyy hh24:mi
    ↵ '), to_date('07-05-2021 17:30', 'dd-mm-yyyy hh24:mi'));
771 |
772 | INSERT INTO ISTORIC_POPASURI
773 | VALUES(SEQ_IST_POP.NEXTVAL, 180, 20120, to_date('05-05-2021 17:00', 'dd-mm-yyyy hh24:mi
    ↵ '), to_date('05-05-2021 17:30', 'dd-mm-yyyy hh24:mi'));
774 |
775 | commit; -- salvam inserarile

```

5 Probleme rezolvate de baza de date

Afisati pentru toti soferi toate popasurile in care s-au oprit in ziua in care s-a oprit pentru prima data un sofer la restaurantul Ceptura din Ploiesti, sa se afiseze daca soferul a venit cu un camion inmatriculat in Prahova sau in alta regiune, numele, prenumele si salariul soferului, numele si tipul popasului la care s-a oprit, la ce ora a sosit, cat a stat, cati bani a cheltuit la popas, si locatia popasului. Locatia sa fie afisata pe o singura coloana numita Locatie popas. Un sofer cheltuieste la un popas in functie de salariul sau, soferii cu un salariu mai mic de 3000lei inclusiv vor cheltui doar 0.005% din salariu, cei cu salariu intre (3000, 4500] vor cheltui 0.01% din salariu, iar restul vor cheltui 0.015% din salariu. Sa se ordoneze dupa nume si dupa prenume. Din lista de angajati nu afisati nimic pentru ultimii doi din lista.

```

01 | -- functie pentru a obtine ziua in care s-a oprit pentru prima data un sofer
02 | -- la restaurantul un popas cu numele si locatia data
03 | CREATE OR REPLACE FUNCTION func_obtine_data(
04 |     p_nume_popas popas.nume%type, p_localitate locatie.localitate%TYPE)
05 | RETURN varchar2
06 |     AS v_data_rez varchar2(255);
07 | BEGIN
08 |     SELECT to_char(min(sip.data_sosire), 'dd-mm-yyyy')
09 |     INTO v_data_rez
10 |     FROM locatie sl JOIN popas sp ON (sl.id_locatie = sp.id_locatie)
11 |             JOIN istoric_popasuri sip ON(sp.id_popas = sip.id_popas)
12 |     WHERE initcap(sp.nume) = p_nume_popas and initcap(sl.localitate) = p_localitate;
13 |
14 |     return v_data_rez;
15 | END;
16 |
17 | -- subprogram pentru a rezolva cerinta data (am ales procedura intrucat nu trebuie
18 | -- sa intoarcem nimic) - folosim o procedura cu 2 parametrii de intrare(default
19 | -- sunt de intrare setati) pentru a rezolva cerinta la un caz mai general (indiferent
20 | -- de popas)
21 | CREATE OR REPLACE PROCEDURE proc_track_sofer(
22 |     p_nume_popas popas.nume%type, p_localitate locatie.localitate%TYPE)
23 | AS
24 |     -- record pentru a retine date despre soferi (folosim tipurile de date din
25 |     -- tabelul de angajati)

```

```

26 |     TYPE rec_sofer_info IS RECORD (
27 |         id_angajat      angajat.id_angajat%TYPE,
28 |         nume            angajat.nume%TYPE,
29 |         prenume          angajat.prenume%TYPE,
30 |         salariu          angajat.salariu%TYPE,
31 |         bani_cheltuiti    angajat.salariu%TYPE
32 |     );
33 |
34 |     -- tabel indexat (INTEGER) cu model de date de tipul record rec_sofer_info
35 |     TYPE t_sofer_info IS TABLE OF rec_sofer_info INDEX BY PLS_INTEGER;
36 |
37 |     -- record pentru tracking al soferului la un popas
38 |     TYPE rec_track_info IS RECORD (
39 |         din_ph    varchar2(255),
40 |         nume_popas popas.nume%TYPE,
41 |         tip_popas popas.tip_popas%TYPE,
42 |         ora_sosire varchar2(255),
43 |         durata    varchar2(255),
44 |         locatie_popas varchar2(255)
45 |     );
46 |     -- tabel imbricat cu model de date de tipul record rec_track_info
47 |     TYPE t_track_info IS TABLE OF rec_track_info;
48 |
49 |     -- tabel indexat in care vom retine soferi in functie de criteriul stabilit
50 |     t_soferi t_sofer_info;
51 |     -- tabel imbricat care va retine date de tracking pt sofer
52 |     t_track t_track_info;
53 |     -- data primei opriri a unui sofer la Restaurantul Ceptura din Ploiesti
54 |     v_data   varchar2(255);
55 |     -- pentru a prinde exceptiile de orice fel (in principiu pentru exceptiile
56 |     -- aplicatiei)
56 |     exceptie EXCEPTION;
57 |     PRAGMA EXCEPTION_INIT(exceptie, -20001);
58 |
59 |     -- cursor clasic pentru a obtine informatii despre toti soferii
60 |     CURSOR c_obtine_soferi IS
61 |         SELECT id_angajat, nume, prenume, salariu,
62 |             CASE WHEN salariu <=3000 THEN salariu * 0.005
63 |                 WHEN salariu <=4500 THEN salariu * 0.01
64 |                 WHEN salariu > 4500 THEN salariu * 0.015
65 |             END
66 |             FROM angajat
67 |             WHERE tip_angajat = 'SOFER' -- vrem doar soferii din toti angajatii
68 |             ORDER BY nume, prenume; -- ordoneaza soferii crescator dupa nume si prenume
69 |
70 |     -- cursor parametrizat pentru a obtine date de tracking despre un sofer dat dupa id
70 |     -- -ul de angajat
71 |     CURSOR c_track(cp_id_angajat angajat.id_angajat%TYPE, cp_data varchar2) IS
72 |         SELECT
73 |             -- decode pentru a verifica nr de inmatriculara al camionului
74 |             -- pe care il
75 |             -- conducea soferul la momentul opririi la popas, primele 2
76 |             -- caractere ne
77 |             -- spun daca a fost inmatriculat in Prahova sau in alta regiune
78 |             DECODE((SELECT upper(substr(sc.nr_inmatriculara, 0, 2))
79 |                     FROM camion sc JOIN istoric_camioane_conduse sic ON(sc.id_camion
80 |                     = sic.id_camion)
81 |                     WHERE sic.id_angajat = cp_id_angajat -- verificam angajatul
82 |                         -- verificam sa obtinem
83 |                         -- camionul pe care il conducea la momentul dat
84 |                         and ip.data_sosire < NVL(sic.data_sfarsit, sysdate)
85 |                         and ip.data_sosire > NVL(sic.data_inceput, sysdate)
86 |                         ), 'PH', 'in Prahova', 'din alta regiune') "Verifica camion din Prahova",
87 |                         p.nume "Nume popas", initcap(p.tip_popas) "Tip popas",
88 |                         to_char(ip.data_sosire, 'hh24:mi') "Ora sosire",
89 |                         substr(numtodsinterval((ip.data_plecare - ip.data_sosire), 'DAY'), 12, 5) "
90 |                         Durata",
90 |                         'Jud: ' || NVL(l.judet, '-') || ' Loc: ' || NVL(l.localitate, '-') ||
90 |                         ' Str. ' || NVL(l.strada, '-') || ' Nr. ' || NVL(to_char(l.nr), '-') "
90 |                         Locatie popas"
91 |             FROM istoric_popasuri ip JOIN popas p ON(ip.id_popas = p.id_popas)
92 |                                         JOIN locatie l ON(p.id_locatie = l.id_locatie)
93 |                                         -- pentru angajatul dat ca parametru si orice popas din ziua data

```

```

91 |         WHERE ip.id_angajat = cp_id_angajat AND to_char(data_sosire, 'dd-mm-yyyy') =
92 |             ↵ cp_data;
93 | BEGIN
94 |     -- obtinem data ceruta din cerinta - folosim parametrii din procedura
95 |     -- si ii pasam functiei
96 |     v_data := func_obtine_data(p_nume_popas, p_localitate);
97 |
98 |     -- daca nu a fost obtinuta o data pentru tracking atunci urmatoarele queries
99 |     -- nu mai au sens
100 |     IF v_data is null THEN
101 |         RAISE_APPLICATION_ERROR(-20001, 'Nu a oprit niciun sofer la locatia data. '
102 |         || 'Daca rezultatul nu este cel asteptat verificati datele de intrare.');
103 |     END IF;
104 |
105 |     -- obtine toti soferii pentru inceput folosind cursorul clasic
106 |     OPEN c_obtine_soferi; -- deschidem cursorul pentru a obtine date
107 |     FETCH c_obtine_soferi BULK COLLECT INTO t_soferi;
108 |     CLOSE c_obtine_soferi; -- dupa ce am obtinut date nu mai avem nevoie de cursor
109 |     -- deci il inchidem, analog pentru cel parametrizat de la liniile 137-139
110 |
111 |     -- din conditia sa nu se ia in considerare ultimii 2 angajati
112 |     -- putem folosi direct COUNT intrucat nu s-au realizat modificari in tabloul
113 |     -- indexat momentan deci ultimul element va fi reprezentat de nr COUNT
114 |     -- dupa prima stergerea penultimul element va fi reprezentat tot de nr COUNT
115 |     -- intrucat al doilea COUNT este diferit de primul COUNT
116 |     -- SECOND_COUNT = FIRST_COUNT - 1;
117 |     t_soferi.DELETE(t_soferi.COUNT);
118 |     t_soferi.DELETE(t_soferi.COUNT);
119 |
120 |     -- simulam cazul in care nu am obtinut soferii
121 |     -- t_soferi.DELETE();
122 |     IF t_soferi.COUNT() = 0 THEN
123 |         RAISE_APPLICATION_ERROR(-20002, 'Nu s-au putut obtine soferii');
124 |     END IF;
125 |
126 |     FOR i IN t_soferi.FIRST..t_soferi.LAST LOOP
127 |         IF t_soferi.EXISTS(i) THEN -- daca exista angajatul
128 |
129 |             -- obtine informatii despre tracking pentru soferul curent din loop
130 |             OPEN c_track(t_soferi(i).id_angajat, v_data);
131 |             FETCH c_track BULK COLLECT INTO t_track;
132 |             CLOSE c_track;
133 |
134 |             -- afiseaza date despre sofer
135 |             DBMS_OUTPUT.PUT_LINE('Soferul '||t_soferi(i).prenume||' '||t_soferi(i).nume||'
136 |             ↵ are salariu de '
137 |             ↵     ||t_soferi(i).salariu||' de lei si cheltuieste in medie '||t_soferi(i).
138 |             ↵ bani_cheltuiti||' lei '
139 |             ↵     ||'la un popas. Informatii de tracking:');
140 |
141 |             -- soferul curent nu s-a oprit la niciun popas in ziua respectiva
142 |             -- nu vrem sa aruncam o exceptie ci sa afisam un mesaj explicit pentru acest
143 |             ↵ caz special
144 |             IF t_track.COUNT() = 0 THEN
145 |                 DBMS_OUTPUT.PUT_LINE('Nu s-a oprit la niciun popas in data de '||v_data);
146 |             ELSE
147 |                 FOR j IN t_track.FIRST..t_track.LAST LOOP
148 |                     DBMS_OUTPUT.PUT_LINE('S-a oprit cu un camion inmatriculat '||t_track(j)
149 |                     ↵ .din_ph||' la un '|| lower(t_track(j).tip_popas)||' din '||t_track(j).locatie_popas||' la
150 |                     ↵ ora '||t_track(j).ora_sosire||' unde a stat '
151 |                     ↵     ||'pentru o perioada de '||t_track(j).durata||' ore.');
152 |                 END LOOP;
153 |             END IF;
154 |
155 |             -- separator soferii
156 |             DBMS_OUTPUT.PUT_LINE('-----');
157 |         END IF;
158 |     t_track.DELETE();
159 | END LOOP;
160 |
161 | EXCEPTION
162 |     WHEN exceptie THEN -- prinde toate exceptiile si afiseaza codul si mesajul

```

```

158 |         DBMS_OUTPUT.PUT_LINE('Cod exceptie: ' || SQLCODE || ' ; Mesaj: ' || SQLERRM);
159 |     END;
160 |
161 | -- testam procedura
162 | DECLARE
163 | BEGIN
164 |     proc_track_sofer('Restaurant Ceptura', 'Ploiesti');
165 | END;
166 | -- varianta cu exceptia nu a gasit data
167 | DECLARE
168 | BEGIN
169 |     proc_track_sofer('Restaurant a', 'Ploiesti');
170 | END;

```

Pentru fiecare camion sa se afiseze marca, nr. de inmatriculare, numele firmei, care il detine si un status de transporturi. Pentru fiecare transport afisati data de plecare, soferul care a realizat transprotul si cu cate este platit lunar. Daca cu un camion nu au fost realizeate transporturi atunci afisati un mesaj corespunzator. Folositi o procedura pentru cerinta de tracking.

```

01 | CREATE OR REPLACE PROCEDURE tracking_proc AS
02 |     TYPE refcursor IS REF CURSOR;
03 |     -- ciclu cursor care obtine date despre un camion si un cursor pentru
04 |     -- transporturile
05 |     -- realizate de el
06 |     CURSOR c_camion IS
07 |         SELECT c.id_camion id_camion, c.marca marca, c.nr_inmatriculare
08 |         -- nr_inmatriculare,
09 |         f.nume nume_detinator,
10 |         CURSOR (WITH ang AS -- soferii angajati inainte de 13 sept 2020 care detin
11 |             cel mult 2 permise auto
12 |                 (SELECT p.id_angajat, a.nume, a.prenume, a.salariu
13 |                  FROM angajat a JOIN permis p ON (a.id_angajat = p.id_angajat)
14 |                  WHERE a.tip_angajat = 'SOFER'
15 |                      and to_char(a.data_angajare, 'dd-mm-yyyy') < '13-9-2020'
16 |                      GROUP BY p.id_angajat, a.nume, a.prenume, a.salariu
17 |                      HAVING count(p.id_angajat) >= 1 -- care detine cel putin 1
18 |             -->
19 |             permis
20 |             )
21 |             SELECT a.nume nume_sofer, a.prenume prenume_sofer, a.salariu
22 |             salariu_sofer,
23 |             to_char(t.data_plecare, 'dd-mm-yyyy') data_transport
24 |             FROM transport t JOIN istoric_camioane_conduse icc ON (t.id_camion
25 |             -->
26 |             = icc.id_camion)
27 |             JOIN ang a ON (icc.id_angajat = a.id_angajat)
28 |             WHERE icc.id_camion = c.id_camion
29 |                 and ((icc.data_inceput < t.data_plecare
30 |                     and t.data_plecare <= NVL(icc.data_sfarsit, sysdate)
31 |                     or t.data_plecare is null)
32 |             )
33 |             FROM camion c JOIN firma f ON (c.id_firma = f.id_firma)
34 |             WHERE upper(c.marca) != 'IVECO' and f.tip_firma = 'TRANSPORT';
35 |
36 |     -- variabile pentru a retine date despre fiecare camion
37 |     v_id_camion          camion.id_camion%TYPE;
38 |     v_marca_camion       camion.marca%TYPE;
39 |     v_nr_inmatriculare_camion camion.nr_inmatriculare%TYPE;
40 |     v_nume_detinator_camion firma.nume%TYPE;
41 |     v_cursor_transporturi refcursor; -- variabila care refera un cursor
42 |
43 |     -- record pentru a retine informatii minimale despre un transport si sofer
44 |     TYPE rec_transport IS RECORD (
45 |         nume_sofer           angajat.nume%TYPE,
46 |         prenume_sofer        angajat.nume%TYPE,
47 |         salariu              angajat.salariu%TYPE,
48 |         data_transport        varchar2(255)
49 |     );
50 |     rt rec_transport; -- variabila de tip record
51 | BEGIN
52 |     OPEN c_camion; -- deschide cursorul principal

```

```

45 |      LOOP
46 |        FETCH c_camion INTO v_id_camion, v_marca_camion, v_nr_inmatriculare_camion,
47 |          v_nume_detinator_camion, v_cursor_transporturi;
48 |        EXIT WHEN c_camion%NOTFOUND; -- nu mai sunt camioane de afisate
49 |        DBMS_OUTPUT.PUT_LINE('-----');
50 |        -- separator
51 |        DBMS_OUTPUT.PUT_LINE('Camionul cu marca'||v_marca_camion||' avand nr. de
52 |        inmatriculare '
53 |          ||v_nr_inmatriculare_camion||' si este detinut de compania '
54 |          ||v_nume_detinator_camion||'.');
55 |        DBMS_OUTPUT.PUT_LINE('Status transporturi:');
56 |
57 |        -- pentru fiecare camion afisam informatiile despre status transporturi
58 |        LOOP
59 |          FETCH v_cursor_transporturi INTO rt; -- obtine informatii
60 |          -- iesi din loop atunci cand nu mai sunt transporturi
61 |          EXIT WHEN v_cursor_transporturi%NOTFOUND;
62 |
63 |          -- afiseaza informatii despre transport si sofer
64 |          DBMS_OUTPUT.PUT_LINE('Transport in data de'||rt.data_transport||' realizat
65 |          de '
66 |            ||rt.nume_sofer||' ||rt.prenume_sofer||' care este platit lunar cu '
67 |            ||rt.salariu||'lei.');
68 |        END LOOP;
69 |
70 |        -- verificam cu rowcount daca nu am gasit date in variabila cursor
71 |        -- semnificand ca nu s-au realizat transporturi cu acest camion inand
72 |        -- cont de criteriile date
73 |        IF v_cursor_transporturi%rowcount = 0 THEN
74 |          DBMS_OUTPUT.PUT_LINE('Nu au fost realizate transporturi cu acest camion.');
75 |        END IF;
76 |
77 |        DBMS_OUTPUT.NEW_LINE(); -- separam cu o linie la final
78 |      END LOOP;
79 |
80 |      -- executa
81 |      BEGIN
82 |        tracking_proc();
83 |      END;

```

Creati o functie care sa intoarca urmatoarele date despre un sofer dat prin 2 parametrii de intrare (numele si prenumele sau): nr. de telefon pentru a putea fi contact, cand a fost angajat, la ce firma lucreaza, ce marca de camion conduce acum, nr. de inmatriculare si cate camioane a condus pana acum. Tratati urmatoarele cazuri prin exceptii:

- nu a fost gasit angajatul
- exista mai multi angajati cu acelasi nume
- angajatul exista dar nu este sofer
- celelalte exceptii care pot aparea generic, cod de eroare si mesajul erorii

```

01 | -- tabel ajutator pentru a intoarce un tip de date din functie
02 | CREATE TABLE tab8_ang_info (
03 |   nr_telefon_ang           varchar2(25),
04 |   data_angajarre           date,
05 |   salariu_angajat          number(10),
06 |   nume_firma                varchar2(15) ,
07 |   marca_camion              varchar2(10),
08 |   nr_inmatriculare          varchar2(10),
09 |   nr_camioane_conduse       number
10 | );
11 |
12 | -- creeaza intr-un pachet eroarea pentru a o putea arunca in functie si a o
13 | -- prinde in blocul in care apelam functia. Facem acest lucru intrucat daca

```

```

14 | -- definim exceptia in functie ea este declarata doar in blocul privat al functiei
15 | -- si daca incercam sa o definim si in blocul apelant ea nu va fi considerata
16 | -- aceeasi si va intra pe cazul others;
17 | CREATE OR REPLACE PACKAGE p8_exceptions
18 | AS
19 |     nu_este_sofer EXCEPTION;
20 |     PRAGMA EXCEPTION_INIT(nu_este_sofer, -20080);
21 | END;
22 |
23 |
24 | CREATE OR REPLACE FUNCTION func8_info_sofer(
25 |     p_nume angajat.nume%TYPE, p_prenume angajat.prenume%TYPE)
26 | RETURN tab8_ang_info%rowtype IS
27 |     v_tip_angajat angajat.tip_angajat%TYPE;
28 |     rezultat tab8_ang_info%rowtype;
29 |     nu_este_sofer EXCEPTION;
30 | BEGIN
31 |     -- verificam de dinainte tipul angajatului sa stim daca nu se vor intoarce
32 |     -- date de la query-ul urmator
33 |     SELECT tip_angajat
34 |     INTO v_tip_angajat
35 |     FROM angajat
36 |     WHERE nume = p_nume AND prenume = p_prenume;
37 |
38 |     IF v_tip_angajat <> 'SOFER' THEN -- angajatul nu este sofer, aruncam exceptie
39 |         RAISE p8_exceptions.nu_este_sofer;
40 |     END IF;
41 |
42 |     -- daca este sofer obtinem datele cerute si le salvam in rezultat
43 |     SELECT a.nr_telefon, a.data_angajare, a.salariu, f.nume, c.marca, c.
44 |     ↪ nr_inmatriculara,
45 |     ↪ (SELECT count(id_camion) -- numaram cate camioane a condus soferul folosind o
46 |     ↪ subcerere
47 |         FROM istoric_camioane_conduse sicc
48 |         WHERE sicc.id_angajat = a.id_angajat) nr_camioane_conduse
49 |     INTO rezultat
50 |     FROM angajat a JOIN istoric_camioane_conduse icc ON (a.id_angajat = icc.id_angajat)
51 |         JOIN camion c ON (icc.id_camion = c.id_camion)
52 |         JOIN firma f ON (a.id_firma = f.id_firma)
53 |     WHERE icc.data_sfarsit is null -- null marcheaza faptul ca inca conduce camionul
54 |         AND a.nume = p_nume AND a.prenume = p_prenume;
55 |
56 |     RETURN rezultat;
57 | END;
58 |
59 | DECLARE
60 |     v_ang tab8_ang_info%rowtype;
61 | BEGIN
62 |     v_ang := func8_info_sofer('Ion', 'Vasile'); -- ex bun
63 |     -- v_ang := func8_info_sofer('Andrei', 'Vasile'); -- no data found
64 |     -- v_ang := func8_info_sofer('Ion', 'Catalin'); -- too many rows
65 |     -- v_ang := func8_info_sofer('Andrei', 'David'); -- exceptie speciala nu este sofer
66 |     -- o simpla afisare a datelor pentru a verifica cazul bun
67 |     DBMS_OUTPUT.PUT_LINE('Soferul poate fi contact la nr'||v_ang.nr_telefon_ang||
68 |     ' Lucreaza pentru firma'||v_ang.nume_firma||' unde primeste salariul de '|
69 |     '||v_ang.salariu_angajat||'. A condus in total '||v_ang.nr_camioane_conduse|
70 |     '||' pana acum. In momentul de fata conduce un camion'||v_ang.marca_camion|
71 |     '||' cu nr. de inmatriculara'||v_ang.nr_inmatriculara);
72 | EXCEPTION
73 |     WHEN NO_DATA_FOUND THEN
74 |         DBMS_OUTPUT.PUT_LINE('Nu au putut fi obtinute informatii despre soferul dat');
75 |     WHEN TOO_MANY_ROWS THEN
76 |         DBMS_OUTPUT.PUT_LINE('Au fost gasiti mai multi angajati cu numele dat');
77 |     WHEN p8_exceptions.nu_este_sofer THEN
78 |         DBMS_OUTPUT.PUT_LINE('Angajatul dat nu este sofer. Nu putem intoarce informatii
79 |         ↪ despre el.');
80 |     WHEN others THEN
81 |         DBMS_OUTPUT.PUT_LINE('Cod eroare: '||SQLCODE);
82 |         DBMS_OUTPUT.PUT_LINE('Mesaj erorare: '|| SQLERRM);
83 | END;

```

Creati o procedura care primeste nr. de inmatriculare al unui camion si o data de plecare, si obtineti urmatoarele informatii despre transportul realizat de camion in ziua respectiva. Se considera ca un camion nu poate realiza mai multe transporturi in aceeasi zi. Afisati urmatoarele informatii despre transport: nr. de inmatriculare, judetul in care a fost inmatriculat, acum cate luni a fost realizat transportul, data de plecare a camionului din depozitul de plecare, cine detine depozitele de plecare si destinatie si locatia acestora. Mai afisati o lista cu tipurile de marfa transportata, numele marfii si cantitatea. Tratati toate exceptiile care pot aparea:

1. nu niciun transport pentru datele oferite
2. camionul a realizat mai multe transporturi in aceeasi zi - invalid nu poate sa fie in 2 locuri in acelasi timp
3. exista transportul, dar nu exista informatii despre ce marfa transporta
4. alte exceptii

```

01 | CREATE OR REPLACE PROCEDURE proc9_afisare_transport(
02 |   p_nr_inmatriculare camion.nr_inmatriculare%TYPE,
03 |   p_data_plecare varchar2
04 | ) AS
05 |   TYPE rec_transport_info IS RECORD (
06 |     id_transport          transport.id_transport%TYPE,
07 |     marca_camion          camion.marca%TYPE,
08 |     nr_inmatriculare      camion.nr_inmatriculare%TYPE,
09 |     judet_inmatriculare   varchar2(255),
10 |     data_ora_plecare      varchar2(255),
11 |     nr_luni_de_la_transport number,
12 |     detinator_depozit_plecare firma.nume%TYPE,
13 |     locatie_depozit_plecare varchar2(255),
14 |     detinator_depozit_destinatie firma.nume%TYPE,
15 |     locatie_depozit_destinatie varchar2(255)
16 | );
17 |   rez rec_transport_info;
18 |   -- pentru a verifica validitatea primul transport;
19 |   rez_test_valid rec_transport_info;
20 |
21 |   -- record pentru a retine fiecare obiect de marfa transportat
22 |   TYPE rec_transport_marfa IS RECORD (
23 |     marfa          lot_marfa.nume%TYPE,
24 |     cantitate     lot_marfa.cantitate%TYPE,
25 |     tip_aliment   varchar2(255)
26 | );
27 |   -- tabel imbricat pentru a salva marfa din transport
28 |   TYPE tab_marfa IS TABLE OF rec_transport_marfa;
29 |   tm tab_marfa := tab_marfa();
30 |
31 |   -- exceptie pentru cazul in care gasim un transport dar nu si marfa acestuia
32 |   -- eroare cauzata din lipsa de inserare a marfii la inserarea transportului
33 |   transport_fara_marfa EXCEPTION;
34 |   PRAGMA EXCEPTION_INIT(transport_fara_marfa, -20090);
35 |
36 |   -- obtine informatiile cerute despre un transport realizat de un camion
37 |   -- dat prin nr. de inmatriculare intr-o zi data
38 |   CURSOR c_transport(
39 |     cp_nr_inmatriculare camion.nr_inmatriculare%TYPE,
40 |     cp_data_plecare    varchar2
41 |   ) IS
42 |     WITH depozit_plecare AS
43 |       (SELECT id_depozit, sf.nume detinator, 'Loc: ' || NVL(sl.localitate, '-')
44 |        || ' Str. ' || NVL(sl.strada, '-') || ' Nr. ' || NVL(to_char(sl.nr), '-'
45 |        ) locatie
46 |         FROM firma sf JOIN depozit sd ON (sf.id_firma = sd.id_firma)
47 |             JOIN locatie sl ON (sd.id_locatie = sl.id_locatie)
48 |         ),
49 |         depozit_destinatie AS
50 |           (SELECT id_depozit, sf.nume detinator, 'Loc: ' || NVL(sl.localitate, '-')
51 |            || ' Str. ' || NVL(sl.strada, '-') || ' Nr. ' || NVL(to_char(sl.nr), '-'
52 |            ) locatie
53 |             FROM firma sf JOIN depozit sd ON (sf.id_firma = sd.id_firma)
54 |                 JOIN locatie sl ON (sd.id_locatie = sl.id_locatie)
55 |             ),
56 |             marfa AS
57 |               (SELECT lot_marfa.nume marfa, lot_marfa.cantitate cantitate
58 |                FROM tab_marfa
59 |               )
60 |             );
61 |   
```

```

50 |           || ' Str. ' || NVL(sl.strada, '-') || ' Nr. ' || NVL(to_char(sl.nr), '-'
51 |           ↵ ) locatie
52 |               FROM firma sf JOIN depozit sd ON (sf.id_firma = sd.id_firma)
53 |                           JOIN locatie sl ON (sd.id_locatie = sl.id_locatie)
54 |           )
55 |           SELECT t.id_transport,
56 |                   c.marca "Marca camion", c.nr_inmatriculare "Nr. inmatriculare",
57 |                   CASE WHEN substr(c.nr_inmatriculare, 0, 2)= 'MM' THEN 'Maramures'
58 |                           WHEN substr(c.nr_inmatriculare, 0, 2)= 'IS' THEN 'Iasi'
59 |                           WHEN substr(c.nr_inmatriculare, 0, 2)= 'CT' THEN 'Constanta'
60 |                           WHEN substr(c.nr_inmatriculare, 0, 1)= 'B' THEN 'Bucuresti'
61 |                           ELSE 'Necunoscut'
62 |                   END "Inmatriculat in judetul",
63 |                   to_char(t.data_plecare, 'dd-mm-yyyy hh24:mi') "Data si ora plecare",
64 |                   round(months_between(sysdate, t.data_plecare)) "Nr. luni de la transport
65 |           ↵ ",
66 |                   dp.detinator "Detinator depozit plecare", dp.locatie "Locatie depozit
67 |           ↵ plecare",
68 |                   dd.detinator "Detinator depozit destinatie", dd.locatie "Locatie depozit
69 |           ↵ destinatie"
70 |               FROM transport t JOIN camion c ON(t.id_camion = c.id_camion)
71 |                           JOIN firma f ON(c.id_firma = f.id_firma)
72 |                           JOIN depozit_plecare dp ON(dp.id_depozit = t.depozit_plecare)
73 |                           JOIN depozit_destinatie dd ON(dd.id_depozit = t.
74 |           ↵ depozit_destinatie)
75 |                   WHERE c.nr_inmatriculare = cp_nr_inmatriculare
76 |                           AND to_char(t.data_plecare, 'dd-mm-yyyy') = cp_data_plecare;
77 |
78 |
79 |
80 |
81 |
82 |
83 |
84 |
85 | BEGIN
86 | -- obtine date despre transport
87 |     OPEN c_transport(p_nr_inmatriculare, p_data_plecare); -- deschide cursor
88 |         ↵ parametrizat
89 |         FETCH c_transport INTO rez; -- obtine transportul
90 |         FETCH c_transport INTO rez_test_valid; -- mai facem un fetch de test
91 |         -- pentru a verifica daca camionul a realizat mai mult de un transport in
92 |         -- acea zi, daca a realizat doar un transport rowcount va ramane 1
93 |         -- nu au fost salvate date nici la primul fetch => nu exista transportul
94 |         -- arunca exceptie
95 |         IF c_transport%rowcount = 0 THEN
96 |             RAISE NO_DATA_FOUND;
97 |         END IF;
98 |         -- a fost obtinut mai mult de un transport, transporturile realizate de
99 |         -- camionul dat in ziua data sunt invalide, arunca exceptie custom
100 |        IF c_transport%rowcount > 1 THEN
101 |            RAISE TOO_MANY_ROWS;
102 |        END IF;
103 |        CLOSE c_transport; -- inchide cursor
104 |
105 | -- obtine marfa
106 |     OPEN c_marfa_transport(rez.id_transport); -- deschide cursor marfa
107 |     FETCH c_marfa_transport BULK COLLECT INTO tm; -- obtine loturile de marfa
108 |     -- si salveaza-le in tabelul imbricat de marfa
109 |
110 |     -- daca nu exista loturi de marfa pentru transportul dat atunci transportul
111 |     -- este invalid, arunca exceptie custom fata de cea no_data_found
112 |     IF c_marfa_transport%rowcount = 0 or tm.COUNT() = 0 THEN
113 |         RAISE transport_fara_marfa;
114 |     END IF;

```

```

115 |
116 |     CLOSE c_marfa_transport; -- inchide cursor
117 |
118 |
119 | -- afisare date cerute
120 |     DBMS_OUTPUT.PUT_LINE('Camionul'||rez.nr_inmatriculare||' inmatriculat in judetul '
121 |                         ||rez.judet_inmatriculare||' a realizat un transport acum '
122 |                         ||rez.nr_luni_de_la_transport||' in data de '
123 |                         ||rez.data_ora_plecare);
124 |     DBMS_OUTPUT.PUT_LINE('Camionul a plecat din depozitul detinut de '
125 |                         ||rez.detinator_depozit_plecare||' din '||rez.locatie_depozit_plecare
126 |                         ||' spre depozitul detinut de '||rez.detinator_depozit_destinatie
127 |                         ||' din '||rez.locatie_depozit_destinatie);
128 |
129 |     DBMS_OUTPUT.PUT_LINE('Marfa transportata de camion:');
130 |     FOR i IN tm.FIRST..tm.LAST LOOP
131 |         DBMS_OUTPUT.PUT_LINE(tm(i).tip_aliment||'-'||
132 |                             tm(i).marfa||'-'||tm(i).cantitate);
133 |     END LOOP;
134 |
135 | EXCEPTION
136 |     WHEN NO_DATA_FOUND THEN -- pentru cazul in care nu am gasit transportul
137 |         DBMS_OUTPUT.PUT_LINE('Camionul'||p_nr_inmatriculare||
138 |                             ' nu a realizat niciun transport in data de '||p_data_plecare);
139 |     WHEN TOO_MANY_ROWS THEN -- pentru cazul cand un sofer a realizat mai multe
140 |         → transporturi intr-o zi
141 |         DBMS_OUTPUT.PUT_LINE('Camionul'||p_nr_inmatriculare||
142 |                             ' a realizat mai multe transporturi in data de '||p_data_plecare
143 |                             ||'. Date eronate.');
144 |     WHEN transport_fara_marfa THEN -- daca am gasit transportul dar nu si marfa
145 |         DBMS_OUTPUT.PUT_LINE('Camionul'||p_nr_inmatriculare||
146 |                             ' a realizat un transport in data de '||p_data_plecare||
147 |                             ' dar nu exista inregistrari pentru marfa transportata. Transport invalid.'
148 |                             );
149 |     WHEN OTHERS THEN -- alte exceptiie neprevazute
150 |         DBMS_OUTPUT.PUT_LINE(SQLERRM);
151 |
152 | -- 10100 id transport pentru a i sterge marfa transportata pentru a forta o eroare
153 | -- CT12FSD nr inmatriculare data plecare 11-05-2021
154 | select * from transport WHERE id_transport = 10100;
155 | select * from camion where id_camion =15;
156 | select * from inventar_transport where id_transport = 10100;
157 | DELETE FROM inventar_transport WHERE id_transport = 10100;
158 |
159 | -- actualizam transportul cu id-ul 10090 setand data de 10010
160 | -- pentru a forta o eroare (evident update-ul nu are sens intrucat acelasi camion
161 | -- nu putea realiza 2 transporturi in locuri diferite in acelasi timp)
162 | UPDATE transport
163 | SET data_plecare = to_date('10-05-2021', 'dd-mm-yyyy')
164 | WHERE id_transport = 10010;
165 | select * from transport where id_camion =25;
166 | select * from camion where id_camion =25;
167 |
168 | rollback; -- anulam modificarile facute
169 |
170 | -- executa procedura
171 | BEGIN
172 | --     proc9_afisare_transport('MM57RXF', '05-05-2021'); -- ex bun
173 | --     proc9_afisare_transport('MM57RXF', '10-05-2021'); -- no data found - nu am gasit
174 | --         → transportul
175 | --     proc9_afisare_transport('CT90MMM', '10-05-2021'); -- too many rows - am gasit mai
176 | --         → multe transporturi
177 | --     proc9_afisare_transport('CT12FSD', '11-05-2021'); -- custom error - am gasit
178 | --         → transportul dar nu si marfa
179 | END;

```

Creati un trigger care sa permita inserari, actualizari si stergeri in tabelul ANGAJAT doar in in zilele de munca in intervalul orar 8 dimineata 6 seara. Triggerul trebuie sa blocheze actualizările de id-uri sau tip angajat. Folositi-va de application error pentru

a trata fiecare caz in parte.

```
01 | CREATE OR REPLACE TRIGGER trig_check_ang
02 |   BEFORE INSERT OR UPDATE OR DELETE ON angajat
03 | BEGIN
04 |   -- verificam sa nu se faca actualizari in weekend (sambata si duminica)
05 |   -- sau in timpul programului de lucru 8-18
06 |   IF to_char(sysdate, 'DY') IN ('SAT', 'SUN')
07 |     OR (TO_CHAR(SYSDATE,'HH24') NOT BETWEEN 8 AND 18)
08 |   THEN
09 |     -- afiseaza un mesaj de eroare diferit in functie de comanda executata
10 |     IF INSERTING THEN
11 |       RAISE_APPLICATION_ERROR(-20111, 'Inserarea in tabelul de angajati '
12 |           || 'este permisa doar in timpul programului de lucru!');
13 |     ELSIF DELETING THEN
14 |       RAISE_APPLICATION_ERROR(-20112, 'Stergerea din tabelul de angajati '
15 |           || 'este permisa doar in timpul programului de lucru!');
16 |     ELSE
17 |       RAISE_APPLICATION_ERROR(-20113, 'Actualizarile in tabelul de '
18 |           || 'angajati sunt permise doar in timpul programului de lucru!');
19 |     END IF;
20 |   END IF;
21 |
22 |   -- id-ul unui angajat nu poate fi schimbat - nu vrem sa facem actualizari
23 |   -- recursive in toate tabelele in care apare ca FK
24 |   IF UPDATING('id_angajat') THEN
25 |     RAISE_APPLICATION_ERROR(-20114, 'Nu se poate schimba id-ul unui angajat!');
26 |   END IF;
27 |   -- nu acceptam sa fie schimbat tipul unui angajat intrucat ar trebui sa
28 |   -- rectificam datele inserate pana la momentul actualizarii (daca inainte
29 |   -- un angajat era sofer si avem date de transport despre el atunci ele ar fi
30 |   -- considerate acum invalide deci sterse) nu vrem aces comportament
31 |   IF UPDATING('tip_angajat') THEN
32 |     RAISE_APPLICATION_ERROR(-20114, 'Nu se poate schimba tipul unui angajat!');
33 |   END IF;
34 | END;
35 |
36 | -- exemplu pentru a testa cazul in care azi ar fi zi din weekend
37 | select to_char(to_date('09-01-2022','dd-mm-yyyy'), 'DY')
38 | from dual;
39 |
40 |
41 | INSERT INTO angajat
42 | VALUES(SEQ_ANG.NEXTVAL, 'Ion', 'Vasile', '0722123456',
43 |        to_date('11-04-2020 12:00', 'dd-mm-yyyy hh24:mi'), 5000, 1100, 'SOFER', null);
44 |
45 | DELETE FROM angajat
46 | WHERE id_angajat = 20000;
47 |
48 | UPDATE angajat
49 | SET salariu = 10000
50 | WHERE id_angajat = 20000;
51 |
52 | UPDATE ANGAJAT
53 | SET id_angajat = 40000
54 | WHERE id_angajat = 20000;
55 |
56 | UPDATE ANGAJAT
57 | SET tip_angajat = 'PAZNIC'
58 | WHERE id_angajat = 20000;
59 |
60 | -- comenzi corect in functie de trigger
61 | INSERT INTO angajat
62 | VALUES(SEQ_ANG.NEXTVAL, 'Ion', 'Vasile', '0722123456',
63 |        to_date('11-04-2020 12:00', 'dd-mm-yyyy hh24:mi'), 5000, 1100, 'SOFER', null);
64 | SELECT * FROM ANGAJAT;
65 | UPDATE angajat
66 | SET salariu = 10000
67 | WHERE id_angajat = 20420;
68 | DELETE FROM ANGAJAT
69 | WHERE id_angajat = 20420;
```

Creati un trigger care atunci cand se va realiza update pentru cheile primare din tabelul firma, i.e. id_firma, va actualiza datele in toate tabelele in care este cheie externa. Pentru teste faceti copii dupa tabelele deja existente.

```

01 | -- facem copii dupa tabele pentru noi triggeri
02 | CREATE TABLE depozit_copy AS SELECT * FROM depozit;
03 | CREATE TABLE camion_copy AS SELECT * FROM camion;
04 | CREATE TABLE angajat_copy AS SELECT * FROM angajat;
05 | CREATE TABLE echipa_paza_copy AS SELECT * FROM echipa_paza;
06 | CREATE TABLE firma_copy AS SELECT * FROM firma;
07 |
08 | -- triger care dupa actualizarea pk din tabelul firma actualizeaza fk id_firma
09 | -- in orice tabel care apare
10 | CREATE OR REPLACE TRIGGER update_pk_firma
11 |   AFTER UPDATE OF id_firma ON firma_copy
12 |   FOR EACH ROW
13 | BEGIN
14 |   UPDATE depozit_copy
15 |   SET id_firma = :NEW.id_firma
16 |   WHERE id_firma = :OLD.id_firma;
17 |
18 |   UPDATE camion_copy
19 |   SET id_firma = :NEW.id_firma
20 |   WHERE id_firma = :OLD.id_firma;
21 |
22 |   UPDATE angajat_copy
23 |   SET id_firma = :NEW.id_firma
24 |   WHERE id_firma = :OLD.id_firma;
25 |
26 |   UPDATE echipa_paza_copy
27 |   SET id_firma = :NEW.id_firma
28 |   WHERE id_firma = :OLD.id_firma;
29 | END;
30 |
31 | -- teste pentru fiecare tip de firma
32 | -- 100 600 1100
33 | UPDATE firma_copy
34 | SET id_firma = 101
35 | WHERE id_firma = 100;
36 |
37 | UPDATE firma_copy
38 | SET id_firma = 601
39 | WHERE id_firma = 600;
40 |
41 | UPDATE firma_copy
42 | SET id_firma = 1101
43 | WHERE id_firma = 1100;
44 |
45 | -- verificari
46 | select * from firma_copy;
47 | select * from depozit;
48 | select * from camion_copy;
49 | select * from angajat_copy;
50 | select * from echipa_paza_copy;

```

Definiti un trigger care sa verifice pentru tabelul istoric_popasuri validitatea datelor. Data de plecare nu poate sa fie mai mica decat data de sosire a unui sofer la un popas. La update nu putem modifica decat cele 2 dati, nicio alta coloana.

```

01 | CREATE OR REPLACE TRIGGER trig_ist_pop
02 |   BEFORE UPDATE OR INSERT ON istoric_popasuri
03 |   FOR EACH ROW
04 | BEGIN
05 |   IF INSERTING THEN
06 |     IF :NEW.data_plecare is not null
07 |       and :NEW.data_sosire >= :NEW.data_plecare THEN
08 |         RAISE_APPLICATION_ERROR(-20100,
09 |           'Data de plecare nu poate fi mai mica decat data de sosire');
10 |     END IF;
11 |   ELSE -- UPDATE

```

```

12 |         IF :NEW.id_istoric_popas <> :OLD.id_istoric_popas THEN
13 |             RAISE_APPLICATION_ERROR(-20101, 'id_istoric_popas nu poate fi modificat.');
14 |         END IF;
15 |         IF :NEW.id_popas <> :OLD.id_popas THEN
16 |             RAISE_APPLICATION_ERROR(-20101, 'id_popas nu poate fi modificat.');
17 |         END IF;
18 |         IF :NEW.id_angajat <> :OLD.id_angajat THEN
19 |             RAISE_APPLICATION_ERROR(-20101, 'id_angajat nu poate fi modificat.');
20 |         END IF;
21 |
22 |         -- s-a actualizat data de plecare
23 |         IF :NEW.data_plecare <> :OLD.data_plecare THEN
24 |             -- s-a actualizat si data de sosire
25 |             IF :OLD.data_sosire <> :NEW.data_sosire THEN
26 |                 IF :NEW.data_sosire > :NEW.data_plecare THEN
27 |                     RAISE_APPLICATION_ERROR(-20100,
28 |                                         'Data de plecare nu poate fi mai mica decat data de sosire');
29 |                 END IF;
30 |             ELSE -- nu s-a actualizat data de sosire
31 |                 IF :OLD.data_sosire > :NEW.data_plecare THEN
32 |                     RAISE_APPLICATION_ERROR(-20100,
33 |                                         'Data de plecare nu poate fi mai mica decat data de sosire');
34 |                 END IF;
35 |             END IF;
36 |         ELSE -- data de plecare a ramas aceeasi
37 |             IF :NEW.data_sosire > :OLD.data_plecare THEN
38 |                 RAISE_APPLICATION_ERROR(-20100,
39 |                                         'Data de plecare nu poate fi mai mica sau decat data de sosire');
40 |             END IF;
41 |         END IF;
42 |
43 |     END IF;
44 | END;
45 |
46 | INSERT INTO istoric_popasuri
47 | VALUES(50, 180, 20000, to_date('04-10-2021', 'dd-mm-yyyy'), to_date('04-10-2021', 'dd-
48 |     ↪ mm-yyyy'));
49 | INSERT INTO istoric_popasuri
50 | VALUES(50, 180, 20000, to_date('04-10-2021', 'dd-mm-yyyy'), to_date('03-10-2021', 'dd-
51 |     ↪ mm-yyyy'));
52 | INSERT INTO istoric_popasuri
53 | VALUES(50, 180, 20000, to_date('04-10-2021', 'dd-mm-yyyy'), to_date('05-10-2021', 'dd-
54 |     ↪ mm-yyyy'));
55 | -- exceptie nu putem actualiza id_istoric_popas
56 | UPDATE istoric_popasuri
57 | SET id_istoric_popas = 15
58 | WHERE data_sosire = to_date('05-10-2021', 'dd-mm-yyyy');
59 |
60 | -- exceptie nu putem actualiza id_popas
61 | UPDATE istoric_popasuri
62 | SET id_popas = 15
63 | WHERE data_sosire = to_date('05-10-2021', 'dd-mm-yyyy');
64 |
65 | -- exceptie nu putem actualiza id_angajat
66 | UPDATE istoric_popasuri
67 | SET id_angajat = 15
68 | WHERE data_sosire = to_date('05-10-2021', 'dd-mm-yyyy');
69 |
70 | -- actualizam doar data de sosire
71 | UPDATE istoric_popasuri
72 | SET data_sosire = to_date('05-10-2021', 'dd-mm-yyyy')
73 | WHERE data_sosire = to_date('05-10-2021', 'dd-mm-yyyy');
74 |
75 | -- actualizam doar data de plecare - aceeasi zi
76 | UPDATE istoric_popasuri
77 | SET data_plecare = to_date('05-10-2021', 'dd-mm-yyyy')
78 | WHERE data_sosire = to_date('05-10-2021', 'dd-mm-yyyy');
79 |
80 | -- ex bun update doar data plecare - data plecare mai mare
81 | UPDATE istoric_popasuri

```

```

82 | SET data_plecare = to_date('06-10-2021', 'dd-mm-yyyy')
83 | WHERE data_sosire = to_date('05-10-2021', 'dd-mm-yyyy');
84 |
85 | -- modificam data de plecare si o facem mai mica decat data de sosire
86 | UPDATE istoric_popasuri
87 | SET data_plecare = to_date('05-10-2021', 'dd-mm-yyyy')
88 | WHERE data_sosire = to_date('05-10-2021', 'dd-mm-yyyy');
89 |
90 | -- ex exceptie update ambele dati
91 | UPDATE istoric_popasuri
92 | SET data_plecare = to_date('05-10-2021', 'dd-mm-yyyy'),
93 |     data_sosire = to_date('10-10-2021', 'dd-mm-yyyy')
94 | WHERE data_sosire = to_date('05-10-2021', 'dd-mm-yyyy');
95 |
96 | -- ex bun update ambele dati
97 | UPDATE istoric_popasuri
98 | SET data_plecare = to_date('06-10-2021', 'dd-mm-yyyy'),
99 |     data_sosire = to_date('05-10-2021', 'dd-mm-yyyy')
100 | WHERE data_sosire = to_date('05-10-2021', 'dd-mm-yyyy');

```

Creati un tabel audit in care sa salvati recorduri de tipul: nume baza de date, user, eveniment realizat, tipul obiectului din dictionar pe care s-a realizat evenimentul, numele obiectului din dictionar, un timestamp pentru cand a fost realizat evenimentul. Tabelul va fi populat folosind un trigger de tip LDD care va obtine datele necesare si le va salva in tabel. Dupa compilarea trigger-ului testati-l pe mai multe tipuri de obiecte.

```

01 | CREATE TABLE audit_user (
02 |     nume_bd          VARCHAR2(50),
03 |     user_logat       VARCHAR2(30),
04 |     eveniment        VARCHAR2(20),
05 |     tip_object_referit  VARCHAR2(30),
06 |     nume_object_referit  VARCHAR2(30),
07 |     data             TIMESTAMP(3)
08 | );
09 |
10 | CREATE OR REPLACE TRIGGER audit_schema
11 |     AFTER CREATE OR DROP OR ALTER ON SCHEMA
12 | BEGIN
13 |     INSERT INTO audit_user
14 |         VALUES (SYS.DATABASE_NAME, SYS.LOGIN_USER,
15 |                 SYS.SYSEVENT, SYS.DICTIONARY_OBJ_TYPE,
16 |                 SYS.DICTIONARY_OBJ_NAME, SYSTIMESTAMP(3));
17 | END;
18 |
19 |
20 | CREATE TABLE test (id varchar2(25));
21 | ALTER TABLE test ADD coloana_noua varchar2(25);
22 |
23 | desc test;
24 |
25 | CREATE INDEX index_test ON test(id);
26 | DROP INDEX index_test;
27 |
28 | DROP table test;
29 |
30 | CREATE OR REPLACE TYPE o IS OBJECT (t varchar(2));
31 | DROP TYPE o;
32 |
33 | select * from audit_user;

```

Creati un pachet care sa contine toate obiectele definite la cerintele anterioare.

```

01 | CREATE OR REPLACE PACKAGE pack_ex13 IS
02 |     -- 6
03 |     FUNCTION func_obtine_data(p_numepopas popas.nume%type,
04 |                                 p_localitate locatie.localitate%TYPE) RETURN varchar2;
05 |
06 |     PROCEDURE proc_track_sofer(p_numepopas popas.nume%type, p_localitate locatie.
07 |                                 ↪ localitate%TYPE);

```

```

07 |
08 | -- 7
09 | PROCEDURE tracking_proc;
10 |
11 | -- 8
12 | FUNCTION func8_info_sofer(p_numa_angajat.nume%TYPE, p_prenume_angajat.prenume%TYPE)
13 |     RETURN tab8_ang_info%rowtype;
14 |
15 | -- 9
16 | PROCEDURE proc9_afisare_transport(p_nr_inmatriculare_camion.nr_inmatriculare%TYPE,
17 |     p_data_plecare varchar2);
18 | END pack_ex13;
19 |
20 |
21 | CREATE OR REPLACE PACKAGE BODY pack_ex13 IS
22 | -- 6
23 | FUNCTION func_obtine_data(
24 |     p_numa_popas popas.nume%type, p_localitate locatie.localitate%TYPE)
25 | RETURN varchar2
26 |     AS v_data_rez varchar2(255);
27 | BEGIN
28 |     SELECT to_char(min(sip.data_sosire), 'dd-mm-yyyy')
29 |     INTO v_data_rez
30 |     FROM locatie sl JOIN popas sp ON (sl.id_locatie = sp.id_locatie)
31 |             JOIN istoric_popasuri sip ON (sp.id_popas = sip.id_popas)
32 |     WHERE initcap(sp.nume) = p_numa_popas AND initcap(sl.localitate) = p_localitate;
33 |
34 |     return v_data_rez;
35 | END;
36 |
37 | PROCEDURE proc_track_sofer(
38 |     p_numa_popas popas.nume%type, p_localitate locatie.localitate%TYPE)
39 | AS
40 |     -- record pentru a retine date despre soferi (folosim tipurile de date din
41 |     -- tabelul de angajati)
42 |     TYPE rec_sofer_info IS RECORD (
43 |         id_angajat      angajat.id_angajat%TYPE,
44 |         nume            angajat.nume%TYPE,
45 |         prenume          angajat.prenume%TYPE,
46 |         salariu          angajat.salariu%TYPE,
47 |         bani_cheltuiti    angajat.salariu%TYPE
48 |     );
49 |
50 |     -- tabel indexat (INTEGER) cu model de date de tipul record rec_sofer_info
51 |     TYPE t_sofer_info IS TABLE OF rec_sofer_info INDEX BY PLS_INTEGER;
52 |
53 |     -- record pentru tracking al soferului la un popas
54 |     TYPE rec_track_info IS RECORD (
55 |         din_ph    varchar2(255),
56 |         nume_popas popas.nume%TYPE,
57 |         tip_popas popas.tip_popas%TYPE,
58 |         ora_sosire varchar2(255),
59 |         durata    varchar2(255),
60 |         locatie_popas varchar2(255)
61 |     );
62 |     -- tabel imbricat cu model de date de tipul record rec_track_info
63 |     TYPE t_track_info IS TABLE OF rec_track_info;
64 |
65 |     -- tabel indexat in care vom retine soferi in functie de criteriul stabilit
66 |     t_soferi t_sofer_info;
67 |     -- tabel imbricat care va retine date de tracking pt sofer
68 |     t_track t_track_info;
69 |     -- data primei opriri a unui sofer la Restaurantul Ceptura din Ploiesti
70 |     v_data    varchar2(255);
71 |     -- pentru a prinde exceptiile de orice fel (in principiu pentru exceptiile
72 |     -- aplicatiei)
73 |     exceptie EXCEPTION;
74 |     PRAGMA EXCEPTION_INIT(exceptie, -20001);
75 |
76 |     -- cursor clasic pentru a obtine informatii despre toti soferii
77 |     CURSOR c_obtine_soferi IS
78 |         SELECT id_angajat, nume, prenume, salariu,
79 |                 CASE WHEN salariu <=3000 THEN salariu * 0.005

```

```

79 |             WHEN salariu <=4500 THEN salariu * 0.01
80 |             WHEN salariu > 4500 THEN salariu * 0.015
81 |         END
82 |     FROM angajat
83 |     WHERE tip_angajat = 'SOFER' -- vrem doar soferii din toti angajatii
84 |     ORDER BY nume, prenume; -- ordoneaza soferii crescator dupa nume si prenume
85 |
86 | -- cursor parametrizat pentru a obtine date de tracking despre un sofer dat dupa id
87 | -- -ul de angajat
88 | CURSOR c_track(cp_id_angajat angajat.id_angajat%TYPE, cp_data varchar2) IS
89 |     SELECT
90 |             -- decode pentru a verifica nr de inmatricularare al camionului
91 |             -- pe care il
92 |             -- conducea soferul la momentul opririi la popas, primele 2
93 |             -- caractere ne
94 |             -- spun daca a fost inmatriculat in Prahova sau in alta regiune
95 |             -- DECODE((SELECT upper(substr(sc.nr_inmatricularare, 0, 2))
96 |             -- FROM camion sc JOIN istoric_camioane_conduse sic ON(sc.id_camion
97 |             -- = sic.id_camion)
98 |             WHERE sic.id_angajat = cp_id_angajat -- verificam angajatul
99 |             -- -- verificam sa obtinem
100 |             -- camionul pe care il conducea la momentul dat
101 |             -- and ip.data_sosire < NVL(sic.data_sfarsit, sysdate)
102 |             -- and ip.data_sosire > NVL(sic.data_inceput, sysdate)
103 |             ), 'PH', 'in Prahova', 'din alta regiune') "Verifica camion din Prahova",
104 |             p.nume "Nume popas", initcap(p.tip_popas) "Tip popas",
105 |             to_char(ip.data_sosire, 'hh24:mi') "Ora sosire",
106 |             substr(numtodsinterval((ip.data_plecare - ip.data_sosire), 'DAY'), 12, 5) "
107 |             Durata",
108 |             'Jud: ' || NVL(l.judet, '-') || ' Loc: ' || NVL(l.localitate, '-') ||
109 |             ' Str. ' || NVL(l.strada, '-') || ' Nr. ' || NVL(to_char(l.nr), '-')
110 |             Locatie popas"
111 |     FROM istoric_popasuri ip JOIN popas p ON(ip.id_popas = p.id_popas)
112 |             JOIN locatie l ON(p.id_locatie = l.id_locatie)
113 |             -- pentru angajatul dat ca parametru si orice popas din ziua data
114 |             WHERE ip.id_angajat = cp_id_angajat AND to_char(data_sosire, 'dd-mm-yyyy') =
115 |             cp_data;
116 | BEGIN
117 |     -- obtinem data ceruta din cerinta - folosim parametrii din procedura
118 |     -- si ii pasam functiei
119 |     v_data := func_obtine_data(p_nume_popas, p_localitate);
120 |     -- daca nu a fost obtinuta o data pentru tracking atunci urmatoarele queries
121 |     -- nu mai au sens
122 |     IF v_data is null THEN
123 |         RAISE_APPLICATION_ERROR(-20001, 'Nu a oprit niciun sofer la locatia data. '
124 |         || 'Daca rezultatul nu este cel asteptat verificati datele de intrare.');
125 |     END IF;
126 |     -- obtine toti soferii pentru inceput folosind cursorul clasic
127 |     OPEN c_obtine_soferi; -- deschidem cursorul pentru a obtine date
128 |     FETCH c_obtine_soferi BULK COLLECT INTO t_soferi;
129 |     CLOSE c_obtine_soferi; -- dupa ce am obtinut date nu mai avem nevoie de cursor
130 |     -- deci il inchidem, analog pentru cel parametrizat de liniile 137-139
131 |     -- din conditia sa nu se ia in considerare ultimii 2 angajati
132 |     -- putem folosi direct COUNT intrucat nu s au realizat modificari in tabloul
133 |     -- indexat momentan deci ultimul element va fi reprezentat de nr COUNT
134 |     -- dupa prima stergerea penultimul element vi fi reprezentat tot de nr COUNT
135 |     -- intrucat al doilea COUNT este diferit de primul COUNT
136 |     -- SECOND_COUNT = FIRST_COUNT - 1;
137 |     t_soferi.DELETE(t_soferi.COUNT);
138 |     t_soferi.DELETE(t_soferi.COUNT);
139 |     -- simulam cazul in care nu am obtinut soferii
140 |     -- t_soferi.DELETE();
141 |     -- IF t_soferi.COUNT() = 0 THEN
142 |     --     RAISE_APPLICATION_ERROR(-20002, 'Nu s-au putut obtine soferii');
143 |     -- END IF;
144 |     FOR i IN t_soferi.FIRST..t_soferi.LAST LOOP
145 |         IF t_soferi.EXISTS(i) THEN -- daca exista angajatul

```

```

144 |      -- obtine informatii despre tracking pentru soferul curent din loop
145 |      OPEN c_track(t_soferi(i).id_angajat, v_data);
146 |      FETCH c_track BULK COLLECT INTO t_track;
147 |      CLOSE c_track;
148 |
149 |      -- afiseaza date despre sofer
150 |      DBMS_OUTPUT.PUT_LINE('Soferul'||t_soferi(i).prenume||' '||t_soferi(i).nume||'
151 |      ↪ are salariul de '
152 |      ↪     ||t_soferi(i).salariu||' de lei si cheltuieste in medie '||t_soferi(i).
153 |      ↪ bani_cheltuiti||' lei '
154 |      ↪     ||'la un popas. Informatii de tracking:');
155 |
156 |      -- soferul curent nu s-a oprit la niciun popas in ziua respectiva
157 |      -- nu vrem sa aruncam o exceptie ci sa afisam un mesaj explicit pentru acest
158 |      ↪ caz special
159 |      IF t_track.COUNT() = 0 THEN
160 |          DBMS_OUTPUT.PUT_LINE('Nu s-a oprit la niciun popas in data de '||v_data);
161 |      ELSE
162 |          FOR j IN t_track.FIRST..t_track.LAST LOOP
163 |              DBMS_OUTPUT.PUT_LINE('S-a oprit cu un camion inmatriculat '||t_track(j)
164 |              ↪ .din_ph||' la un '|
165 |              ↪     lower(t_track(j).tip_popas)||' din '||t_track(j).locatie_popas||' la '
166 |              ↪     ora '||t_track(j).ora_sosire||' unde a stat '
167 |              ↪     ||'pentru o perioada de '||t_track(j).durata||' ore.');
168 |          END LOOP;
169 |      END IF;
170 |
171 |      -- separator soferii
172 |      DBMS_OUTPUT.PUT_LINE('-----');
173 |  END IF;
174 |  t_track.DELETE();
175 |END LOOP;
176 |
177 |-- 7
178 |PROCEDURE tracking_proc AS
179 |    TYPE refcursor IS REF CURSOR;
180 |    -- ciclu cursor care obtine date despre un camion si un cursor pentru
181 |    -- transporturile
182 |    -- realizate de el
183 |    CURSOR c_camion IS
184 |        SELECT c.id_camion id_camion, c.marca marca, c.nr_inmatriculare
185 |        ↪ nr_inmatriculare,
186 |        ↪ f.nume nume_detinator,
187 |        -- CURSOR (WITH ang AS -- soferii angajati inainte de 13 sept 2020 care detin
188 |        -- cel mult 2 permise auto
189 |        ↪ (SELECT p.id_angajat, a.nume, a.prenume, a.salariu
190 |        ↪     FROM angajat a JOIN permis p ON (a.id_angajat = p.id_angajat)
191 |        ↪     WHERE a.tip_angajat = 'SOFER'
192 |        ↪         and to_char(a.data_angajare, 'dd-mm-yyyy') < '13-9-2020'
193 |        ↪         GROUP BY p.id_angajat, a.nume, a.prenume, a.salariu
194 |        ↪         HAVING count(p.id_angajat) >= 1 -- care detine cel putin 1
195 |        ↪ permis
196 |        ↪ )
197 |        ↪     SELECT a.nume nume_sofer, a.prenume prenume_sofer, a.salariu
198 |        ↪ salariu_sofer,
199 |        ↪         to_char(t.data_plecare, 'dd-mm-yyyy') data_transport
200 |        ↪     FROM transport t JOIN istoric_camioane_conduse icc ON (t.id_camion
201 |        ↪ = icc.id_camion)
202 |        ↪     JOIN ang a ON (icc.id_angajat = a.id_angajat)
203 |        ↪     WHERE icc.id_camion = c.id_camion
204 |        ↪         and ((icc.data_inceput < t.data_plecare
205 |        ↪             and t.data_plecare <= NVL(icc.data_sfarsit, sysdate)
206 |        ↪             or t.data_plecare is null)
207 |        ↪ )

```

```

206 |      v_id_camion           camion.id_camion%TYPE;
207 |      v_marca_camion        camion.marca%TYPE;
208 |      v_nr_inmatriculare_camion camion.nr_inmatriculare%TYPE;
209 |      v_numar_detinator_camion firma.numar%TYPE;
210 |      v_cursor_transporturi refcursor; -- variabila care refera un cursor
211 |
212 |      -- record pentru a retine informatii minimale despre un transport si sofer
213 |      TYPE rec_transport IS RECORD (
214 |          numar_sofer      angajat.numar%TYPE,
215 |          prenume_sofer    angajat.prenume%TYPE,
216 |          salariu         angajat.salariu%TYPE,
217 |          data_transport   varchar2(255)
218 |      );
219 |      rt rec_transport; -- variabila de tip record
220 |  BEGIN
221 |      OPEN c_camion; -- deschide cursorul principal
222 |      LOOP
223 |          FETCH c_camion INTO v_id_camion, v_marca_camion, v_nr_inmatriculare_camion,
224 |                         v_numar_detinator_camion, v_cursor_transporturi;
225 |          EXIT WHEN c_camion%NOTFOUND; -- nu mai sunt camioane de afisate
226 |          DBMS_OUTPUT.PUT_LINE('-----');
227 |          -- separator
228 |          -- afisam informatiile despre camion
229 |          DBMS_OUTPUT.PUT_LINE('Camionul cu marca'||v_marca_camion||' avand nr. de
230 |          inmatriculare '
231 |              ||v_nr_inmatriculare_camion||' si este detinut de compania '
232 |              ||v_numar_detinator_camion||'.');
233 |          DBMS_OUTPUT.PUT_LINE('Status transporturi:');
234 |
235 |          -- pentru fiecare camion afisam informatiile despre status transporturi
236 |          LOOP
237 |              FETCH v_cursor_transporturi INTO rt; -- obtine informatii
238 |              -- iesi din loop atunci cand nu mai sunt transporturi
239 |              EXIT WHEN v_cursor_transporturi%NOTFOUND;
240 |
241 |              -- afiseaza informatii despre transport si sofer
242 |              DBMS_OUTPUT.PUT_LINE('Transport in data de'||rt.data_transport||' realizat
243 |              de '
244 |                  ||rt.numar_sofer||' ||rt.prenume_sofer||' care este platit lunar cu '
245 |                  ||rt.salariu||'lei.');
246 |          END LOOP;
247 |
248 |          -- verificam cu rowcount daca nu am gasit date in variabila cursor
249 |          -- semnificant ca nu s-au realizat transporturi cu acest camion inand
250 |          -- cont de criteriile date
251 |          IF v_cursor_transporturi%rowcount = 0 THEN
252 |              DBMS_OUTPUT.PUT_LINE('Nu au fost realizate transporturi cu acest camion.');
253 |          END IF;
254 |
255 |          DBMS_OUTPUT.NEW_LINE(); -- separam cu o linie la final
256 |      END LOOP;
257 |      CLOSE c_camion; -- inchidem cursorul
258 |  END;
259 |
260 |  -- 8
261 |  FUNCTION func8_info_sofer(
262 |      p_numar_angajat.numar%TYPE, p_prenume_angajat.prenume%TYPE)
263 |  RETURN tab8_ang_info%rowtype IS
264 |      v_tip_angajat angajat.tip_angajat%TYPE;
265 |      rezultat tab8_ang_info%rowtype;
266 |      nu_ește_sofer EXCEPTION;
267 |  BEGIN
268 |      -- verificam de dinainte tipul angajatului sa stim daca nu se vor intoarce
269 |      -- date de la query-ul urmator
270 |      SELECT tip_angajat
271 |      INTO v_tip_angajat
272 |      FROM angajat
273 |      WHERE numar = p_numar AND prenume = p_prenume;
274 |
275 |      IF v_tip_angajat <> 'SOFER' THEN -- angajatul nu este sofer, aruncam exceptie
276 |          RAISE p8_exceptions.nu_ește_sofer;
277 |      END IF;

```

```

276 |      -- daca este sofer obtinem datele cerute si le salvam in rezultat
277 |      SELECT a.nr_telefon, a.data_angajare, a.salariu, f.nume, c.marca, c.
278 |      ↵ nr_inmatricularare,
279 |      ↵     (SELECT count(id_camion) -- numaram cate camioane a condus soferul folosind o
280 |      ↵     subcerere
281 |      ↵     FROM istoric_camioane_conduse sicc
282 |      ↵     WHERE sicc.id_angajat = a.id_angajat) nr_camioane_conduse
283 |      INTO rezultat
284 |      FROM angajat a JOIN istoric_camioane_conduse icc ON (a.id_angajat = icc.id_angajat)
285 |      JOIN camion c ON (icc.id_camion = c.id_camion)
286 |      JOIN firma f ON (a.id_firma = f.id_firma)
287 |      WHERE icc.data_sfarsit is null -- null marcheaza faptul ca inca conduce camionul
288 |      and a.nume = p_nume and a.prenume = p_prenume;
289 |
290 |
291 |      RETURN rezultat;
292 |  END;
293 |
294 |  -- 9
295 |  PROCEDURE proc9_afisare_transport(
296 |      p_nr_inmatricularare camion.nr_inmatricularare%TYPE,
297 |      p_data_plecare varchar2
298 |  ) AS
299 |      TYPE rec_transport_info IS RECORD (
300 |          id_transport          transport.id_transport%TYPE,
301 |          marca_camion         camion.marca%TYPE,
302 |          nr_inmatricularare   camion.nr_inmatricularare%TYPE,
303 |          judet_inmatricularare varchar2(255),
304 |          data_ora_plecare     varchar2(255),
305 |          nr_luni_de_la_transport number,
306 |          detinator_depozit_plecare firma.nume%TYPE,
307 |          locatie_depozit_plecare varchar2(255),
308 |          detinator_depozit_destinatie firma.nume%TYPE,
309 |          locatie_depozit_destinatie varchar2(255)
310 |      );
311 |      rez rec_transport_info;
312 |      -- pentru a verifica validitatea primul transport;
313 |      rez_test_valid rec_transport_info;
314 |
315 |      -- record pentru a retina fiecare obiect de marfa transportat
316 |      TYPE rec_transport_marfa IS RECORD (
317 |          marfa          lot_marfa.nume%TYPE,
318 |          cantitate     lot_marfa.cantitate%TYPE,
319 |          tip_aliment   varchar2(255)
320 |      );
321 |      -- tabel imbricat pentru a salva marfa din transport
322 |      TYPE tab_marfa IS TABLE OF rec_transport_marfa;
323 |      tm tab_marfa := tab_marfa();
324 |
325 |      -- exceptie pentru cazul in care gasim un transport dar nu si marfa acestuia
326 |      -- eroare cauzata din lipsa de inserare a marfii la inserarea transportului
327 |      -- transport_fara_marfa EXCEPTION;
328 |      PRAGMA EXCEPTION_INIT(transport_fara_marfa, -20090);
329 |
330 |      -- obtine informatiile cerute despre un transport realizat de un camion
331 |      -- dat prin nr. de inmatricularare intr-o zi data
332 |      CURSOR c_transport(
333 |          cp_nr_inmatricularare camion.nr_inmatricularare%TYPE,
334 |          cp_data_plecare      varchar2
335 |      ) IS
336 |          WITH depozit_plecare AS
337 |              (SELECT id_depozit, sf.nume detinator, 'Loc: ' || NVL(sl.localitate, '-')
338 |               || ' Str. ' || NVL(sl.strada, '-') || ' Nr. ' || NVL(to_char(sl.nr), '-'
339 |               ↵ ) locatie
340 |                  FROM firma sf JOIN depozit sd ON (sf.id_firma = sd.id_firma)
341 |                  JOIN locatie sl ON (sd.id_locatie = sl.id_locatie)
342 |                  ),
343 |                  depozit_destinatie AS
344 |                      (SELECT id_depozit, sf.nume detinator, 'Loc: ' || NVL(sl.localitate, '-')

```

```

345 |     SELECT t.id_transport,
346 |         c.marca "Marca camion", c.nr_inmatriculare "Nr. inmatriculare",
347 |         CASE WHEN substr(c.nr_inmatriculare, 0, 2)= 'MM' THEN 'Maramures'
348 |             WHEN substr(c.nr_inmatriculare, 0, 2)= 'IS' THEN 'Iasi'
349 |             WHEN substr(c.nr_inmatriculare, 0, 2)= 'CT' THEN 'Constanta'
350 |             WHEN substr(c.nr_inmatriculare, 0, 1)= 'B' THEN 'Bucuresti'
351 |             ELSE 'Necunoscut'
352 |         END "Inmatriculat in judetul",
353 |         to_char(t.data_plecare, 'dd-mm-yyyy hh24:mi') "Data si ora plecare",
354 |         round(months_between(sysdate, t.data_plecare)) "Nr. luni de la transport
355 |         ",
356 |         dp.detinator "Detinator depozit plecare", dp.locatie "Locatie depozit
357 |         plecare",
358 |         dd.detinator "Detinator depozit destinatie", dd.locatie "Locatie depozit
359 |         destinatie"
360 |     FROM transport t JOIN camion c ON(t.id_camion = c.id_camion)
361 |         JOIN firma f ON(c.id_firma = f.id_firma)
362 |         JOIN depozit_plecare dp ON(dp.id_depozit = t.depozit_plecare)
363 |         JOIN depozit_destinatie dd ON(dd.id_depozit = t.
364 |         depozit_destinatie)
365 |     WHERE c.nr_inmatriculare = cp_nr_inmatriculare
366 |         AND to_char(t.data_plecare, 'dd-mm-yyyy') = cp_data_plecare;
367 |
368 |
369 |
370 |
371 |
372 |
373 |
374 |
375 |
376 | BEGIN
377 | -- obtine date despre transport
378 | OPEN c_transport(p_nr_inmatriculare, p_data_plecare); -- deschide cursor
379 |     -- parametrizat
380 |     FETCH c_transport INTO rez; -- obtine transportul
381 |     FETCH c_transport INTO rez_test_valid; -- mai facem un fetch de test
382 |     -- pentru a verifica daca camionul a realizat mai mult de un transport in
383 |     -- acea zi, daca a realizat doar un transport rowcount va ramane 1
384 |     -- nu au fost salvate date nici la primul fetch => nu exista transportul
385 |     -- arunca exceptie
386 | IF c_transport%rowcount = 0 THEN
387 |     RAISE NO_DATA_FOUND;
388 | END IF;
389 |     -- a fost obtinut mai mult de un transport, transporturile realizate de
390 |     -- camionul dat in ziua data sunt invalide, arunca exceptie custom
391 | IF c_transport%rowcount > 1 THEN
392 |     RAISE TOO_MANY_ROWS;
393 | END IF;
394 | CLOSE c_transport; -- inchide cursor
395 |
396 | -- obtine marfa
397 | OPEN c_marfa_transport(rez.id_transport); -- deschide cursor marfa
398 |     FETCH c_marfa_transport BULK COLLECT INTO tm; -- obtine loturile de marfa
399 |     -- si salveaza-le in tabelul imbricat de marfa
400 |
401 |     -- daca nu exista loturi de marfa pentru transportul dat atunci transportul
402 |     -- este invalid, arunca exceptie custom fata de cea no_data_found
403 | IF c_marfa_transport%rowcount = 0 or tm.COUNT() = 0 THEN
404 |     RAISE transport_fara_marfa;
405 | END IF;
406 |
407 | CLOSE c_marfa_transport; -- inchide cursor
408 |
409 |
410 | -- afisare date cerute

```

```

411 |     DBMS_OUTPUT.PUT_LINE('Camionul '||rez.nr_inmatriculare||' inmatriculat in judetul '
412 |         ||rez.judet_inmatriculare||' a realizat un transport acum '
413 |         ||rez.nr_luni_de_la_transport||' in data de '
414 |         ||rez.data_ora_plecare);
415 |     DBMS_OUTPUT.PUT_LINE('Camionul a plecat din depozitul detinut de '
416 |         ||rez.detinator_depozit_plecare||' din '||rez.locatie_depozit_plecare
417 |         ||' spre depozitul detinut de '||rez.detinator_depozit_destinatie
418 |         ||' din '||rez.locatie_depozit_destinatie);
419 |
420 |     DBMS_OUTPUT.PUT_LINE('Marfa transportata de camion:');
421 |     FOR i IN tm.FIRST..tm.LAST LOOP
422 |         DBMS_OUTPUT.PUT_LINE(tm(i).tip_aliment||'-'||
423 |             tm(i).marfa||'-'||tm(i).cantitate);
424 |     END LOOP;
425 |
426 | EXCEPTION
427 |     WHEN NO_DATA_FOUND THEN -- pentru cazul in care nu am gasit transportul
428 |         DBMS_OUTPUT.PUT_LINE('Camionul '||p_nr_inmatriculare||
429 |             ' nu a realizat niciun transport in data de '||p_data_plecare);
430 |     WHEN TOO_MANY_ROWS THEN -- pentru cazul cand un sofer a realizat mai multe
431 |         → transporturi intr-o zi
432 |         DBMS_OUTPUT.PUT_LINE('Camionul '||p_nr_inmatriculare||
433 |             ' a realizat mai multe transporturi in data de '||p_data_plecare
434 |             ||'. Date eronate.');
435 |     WHEN transport_fara_marfa THEN -- daca am gasit transportul dar nu si marfa
436 |         DBMS_OUTPUT.PUT_LINE('Camionul '||p_nr_inmatriculare||
437 |             ' a realizat un transport in data de '||p_data_plecare||
438 |             ' dar nu exista inregistrari pentru marfa transportata. Transport invalid.'
439 |             → );
440 |     WHEN OTHERS THEN -- alte exceptii neprevazute
441 |         DBMS_OUTPUT.PUT_LINE(SQLERRM);
442 |     END;
443 | END pack_ex13;
444 | /
445 | -- testam o procedura din pachet
EXECUTE pack_ex13.tracking_proc;

```

Realizati inca un pachet cu urmatoarele cerinte:

1. Sa se afiseze numele, prenumele, ce salariu isi doreste, cat de bine este platit, vechimea la locul de munca curent, la ce firma lucreaza, si ce depozit pazesc toti paznicii care lucreaza in depozite din alte localitati fata de cea in care se afla cel mai vizitat popas de soferii de tir si doar pentru paznicii pentru care a intrat sub paza lor in depozit cel putin un transport de marfa. Daca un paznic are salariul mai mic decat media salariul din firma in care lucreaza, atunci el isi doreste o marire de salariu cu 200 lei, in caz contra doresc o marire de 100 lei. Consideram ca un paznic este bine platit daca are un salariu mai mare decat 3000 lei inclusiv. Suma de 3000lei reprezinta media salariilor pazniciilor din Romania inidiferent de locatia pazita(nu intra doar depozitele din gestiunea acestei baze de date). Sa se ordoneze descrescator dupa salariu si in caz de egalitate crescator dupa nume si prenume. Separati problema principală in cat mai multe generale pentru cazul in care mai pot fi refolosite si in alte cerinte. Tratati exceptiile care pot aparea. Daca in baza de date nu sunt la momentul rularii date suficiente aduceti mici modificari pentru a forta erorile pentru a demonstra ca se executa corect in cazul in care vor aparea.
2. Intr-un subprogram sa se afiseze numele, prenumele, nr. de telefon la care poate fi contact soferul, unde este angajat si data de sosire si plecare pentru oprirea sa la cel mai popular popas. Aceeasi cerinta in aceleasi subprogram pentru toate motelurile.

Executati subprogramele din pachet pentru rezolvarea cerintelor.

```

01 | CREATE OR REPLACE PACKAGE pack_ex14 IS
02 |   -- creeaza record pt primul tip
03 |   TYPE rec14_info_ang IS RECORD (
04 |     nume_angajat      varchar2(25),
05 |     prenume_angajat   varchar2(25),
06 |     salariu_actual    number(10),
07 |     salariu_dorit    number(10),
08 |     este_platit_bine  varchar2(255), -- fata de restul de angajati din firma
09 |     vechime_angajat   varchar2(255), -- vechime angajat in firma curenta
10 |     nume_firma        varchar2(25),
11 |     id_echipa_paza   number(5)
12 |   );
13 |
14 |   -- informatii despre depozitul in care lucreaza un paznic
15 |   TYPE rec14_info_dep IS RECORD(
16 |     id_locatie       number(10),
17 |     judet            varchar2(25),
18 |     localitate       varchar2(25),
19 |     strada           varchar2(25),
20 |     nr               number(5),
21 |     nr_colegi        number(5) -- nr colegi ai paznicului
22 |   );
23 |
24 | PROCEDURE obtine_cel_mai_vizitat_popas(p_id_popas OUT popas.id_popas%TYPE);
25 | FUNCTION locatie_popas(p_id_popas popas.id_popas%TYPE) RETURN locatie%rowtype;
26 | PROCEDURE info_salariu_din_firma(
27 |   p_id_firma        IN firma.id_firma%TYPE,
28 |   p_tip_angajat    IN angajat.tip_angajat%TYPE,
29 |   p_medie_salariu OUT angajat.salariu%TYPE,
30 |   p_minim_salariu OUT angajat.salariu%TYPE,
31 |   p_maxim_salariu OUT angajat.salariu%TYPE);
32 | FUNCTION f14_paznic_info(
33 |   p_id_angajat angajat.id_angajat%TYPE,
34 |   p_id_firma    firma.id_firma%TYPE) RETURN rec14_info_ang;
35 | FUNCTION f14_depozit_info(p_id_echipa_paza echipa_paza.id_echipa_paza%TYPE)
36 |   RETURN rec14_info_dep;
37 | PROCEDURE p14_info_cer1;
38 | PROCEDURE p14_info_cer2;
39 | END pack_ex14;
40 |
41 |
42 |
43 | CREATE OR REPLACE PACKAGE BODY pack_ex14 IS
44 |   -- functie pentru a obtine id-ul celui mai vizitat popas la un moment dat
45 | PROCEDURE obtine_cel_mai_vizitat_popas
46 |   (p_id_popas OUT popas.id_popas%TYPE) IS
47 |   exceptie_not_found EXCEPTION;
48 | BEGIN
49 |   SELECT id_popas popas
50 |   INTO p_id_popas
51 |   FROM istoric_popasuri
52 |   GROUP BY id_popas
53 |   HAVING count(id_popas) = (SELECT max(count(id_popas))
54 |                               FROM istoric_popasuri
55 |                               GROUP BY id_popas
56 |                               ); -- pt eroare not found +2;
57 |   -- forteaza eroare pentru too-many
58 |   -- or count(id_popas) = (SELECT min(count(id_popas))
59 |   --                           FROM istoric_popasuri
60 |   --                           GROUP BY id_popas
61 |   --                           );
62 |   -- daca nu a gasit nimic arunca o eroare mai detaliata
63 |   IF SQL%NOTFOUND THEN
64 |     RAISE exceptie_not_found;
65 |   END IF;
66 | EXCEPTION
67 |   WHEN exceptie_not_found THEN
68 |     RAISE_APPLICATION_ERROR(-20141, 'Nu a putut fi gasit cel mai popular popas!');
69 |   WHEN TOO_MANY_ROWS THEN
70 |     RAISE_APPLICATION_ERROR(-20142, 'Au fost gasite mai multe popasuri la fel de
    ↵ vizitate!');");

```

```

71 |      WHEN OTHERS THEN
72 |          RAISE_APPLICATION_ERROR(-20141, 'Nu a putut fi gasit cel mai popular popas!');
73 |
74 |
75 |      ---- pentru a testa exceptiile fortate de la procedura anterioara
76 |      --DECLARE
77 |      --      p_popas.id_popas%TYPE;
78 |      --BEGIN
79 |      --          obtine_cel_mai_vizitat_popas(p);
80 |      --          dbms_output.put_line(p);
81 |      --END;
82 |
83 |      -- functie care primeste id-ul unui popas si intoarce toate datele despre
84 |      -- locatia in care se afla
85 |      FUNCTION locatie_popas(p_id_popas popas.id_popas%TYPE)
86 |          RETURN locatie%rowtype IS
87 |          rez_loc locatie%rowtype;
88 |          exceptie_not_found EXCEPTION;
89 |      BEGIN
90 |          SELECT 1.*
91 |          INTO rez_loc
92 |          FROM locatie l JOIN popas p ON (l.id_locatie = p.id_locatie)
93 |          WHERE p.id_popas = p_id_popas;
94 |
95 |          return rez_loc;
96 |      EXCEPTION
97 |          WHEN NO_DATA_FOUND THEN
98 |              RAISE_APPLICATION_ERROR(-20143, 'Nu a putut fi gasita locatia popasului.');
99 |              RETURN NULL;
100 |
101 |
102 |      ---- test exceptie daca se trimite parametru gresit exemplu anterior
103 |      --DECLARE
104 |      --      l_locatie%rowtype;
105 |      --BEGIN
106 |      --          l := locatie_popas(210);
107 |      --EXCEPTION
108 |      --          WHEN others THEN
109 |          DBMS_OUTPUT.PUT_LINE(SQLERRM);
110 |
111 |
112 |      -- procedura pentru a obtine informatii despre salariile dintr-o firma pentru un
113 |      -- anumit tip de angajati (PAZNIC, SOFER)
114 |      PROCEDURE info_salariu_din_firma(
115 |          p_id_firma      IN firma.id_firma%TYPE,
116 |          p_tip_angajat   IN angajat.tip_angajat%TYPE,
117 |          p_medie_salariu OUT angajat.salariu%TYPE,
118 |          p_minim_salariu OUT angajat.salariu%TYPE,
119 |          p_maxim_salariu OUT angajat.salariu%TYPE) IS
120 |          exceptie EXCEPTION;
121 |      BEGIN
122 |          SELECT round(avg(salariu)), min(salariu), max(salariu)
123 |          INTO p_medie_salariu, p_minim_salariu, p_maxim_salariu
124 |          FROM angajat
125 |          WHERE id_firma = p_id_firma and tip_angajat = p_tip_angajat;
126 |      END;
127 |
128 |      -- obtine informatii despre un paznic
129 |      FUNCTION f14_paznic_info(
130 |          p_id_angajat angajat.id_angajat%TYPE,
131 |          p_id_firma    firma.id_firma%TYPE)
132 |          RETURN rec14_info_ang
133 |          IS resultat rec14_info_ang;
134 |          v_min_salariu    angajat.salariu%TYPE;
135 |          v_medie_salariu  angajat.salariu%TYPE;
136 |          v_max_salariu    angajat.salariu%TYPE;
137 |      BEGIN
138 |          -- obtine medie salariu - restul de date le vom ignora
139 |          info_salariu_din_firma(p_id_firma, 'PAZNIC', v_medie_salariu, v_min_salariu,
140 |          ↪ v_max_salariu);
141 |          IF v_medie_salariu is null THEN -- nu au fost obtinute date, arunca exceptie
142 |              RAISE_APPLICATION_ERROR(-20144, 'Nu au putut fi obtinute informatii despre
143 |              ↪ firma');

```

```

142 |     END IF;
143 |
144 |     -- obtine informatiile si salveaza-le intr-un record
145 |     SELECT a.nume, a.prenume, a.salariu,
146 |         CASE WHEN a.salariu < v_medie_salariu THEN a.salariu + 200
147 |             ELSE a.salariu + 100
148 |         END, -- salariu dorit
149 |         DECODE((a.salariu-3000)-abs(a.salariu-3000)
150 |             , 0, 'bine platit', 'prost platit'), -- este bine platit fata de restul
151 |             concat(substr(numtodsinterval((sysdate - a.data_angajare), 'DAY'), 8, 3) ||
152 |             ' zile ',
153 |                 substr(numtodsinterval((sysdate - a.data_angajare), 'DAY'), 12, 5)), --
154 |             'vechime la locul de munca in firma
155 |                 f.nume, --nume firma,
156 |                 a.id_echipa_paza -- din ce echipa de paza face parte
157 |             INTO rezultat -- salveaza in recordul rezultat
158 |             FROM angajat a JOIN firma f ON (a.id_firma = f.id_firma)
159 |             WHERE a.id_angajat = p_id_angajat
160 |             ORDER BY a.salariu desc, a.nume, a.prenume;
161 |
162 |     RETURN rezultat;
163 | EXCEPTION
164 |     WHEN NO_DATA_FOUND THEN
165 |         RAISE_APPLICATION_ERROR(-20145, 'Nu au fost gasite informatii despre angajat.
166 |             Angajatul nu exista!');
167 |         RETURN NULL;
168 |     END;
169 |
170 |     -- obtine informatiile despre un depozit la care este repartizata o echipa de
171 |     -- paza - informatie despre locatie si nr. de colegi pe care ii are orice paznic
172 |     -- i.e. nr. paznici - 1
173 | FUNCTION f14_depozit_info(p_id_echipa_paza echipa_paza.id_echipa_paza%TYPE)
174 |     RETURN rec14_info_dep;
175 | IS rezultat rec14_info_dep;
176 | BEGIN
177 |     SELECT l.id_locatie, l.judet, l.localitate, l.strada, l.nr,
178 |         (SELECT count(id_echipa_paza)-1
179 |             FROM angajat
180 |             WHERE id_echipa_paza = d.id_echipa_paza
181 |             GROUP BY id_echipa_paza)
182 |             INTO rezultat
183 |             FROM depozit d JOIN locatie l ON (d.id_locatie = l.id_locatie)
184 |             WHERE d.id_echipa_paza = p_id_echipa_paza;
185 |
186 |     RETURN rezultat;
187 | EXCEPTION
188 |     WHEN NO_DATA_FOUND THEN
189 |         RAISE_APPLICATION_ERROR(-20146, 'Nu au fost gasite informatii despre depozitul
190 |             in care este repartizata echipa de paza data!');
191 |     END;
192 |
193 |     ---- testeaza exceptia de la subprogramul anterior
194 |     --DECLARE
195 |     -- rezultat rec14_info_dep;
196 |     --BEGIN
197 |     --     rezultat := f14_depozit_info(2110);
198 |     --EXCEPTION
199 |     --     WHEN OTHERS THEN
200 |     --         DBMS_OUTPUT.PUT_LINE(SQLERRM);
201 |     --END;
202 |
203 |     -- program care le executa
204 | PROCEDURE p14_info_cer1 IS
205 |     -- tabel imbricat pentru informatii despre paznic
206 |     TYPE t_ang_info IS TABLE OF rec14_info_ang;
207 |     ta t_ang_info := t_ang_info();
208 |
209 |     -- tabel imbricat pentru informatii despre depozitele in care lucreaza paznicii
210 |     TYPE t_dep_paz IS TABLE OF rec14_info_dep;
211 |     td t_dep_paz := t_dep_paz();
212 |
213 |     -- record pentru a obtine idurile angajatilor si firmelor la care lucreaza

```

```

211 |      TYPE r_ang_firma IS RECORD (
212 |          id_angajat angajat.id_angajat%TYPE,
213 |          id_firma     firma.id_firma%TYPE
214 |      );
215 |      -- tabel indexat de tip record pentru a se putea face loop cu datele din el
216 |      -- pentru a insera date in tabelele definite anterior folosind functiile
217 |      -- si procedurile create
218 |      TYPE t_ang_firma IS TABLE OF r_ang_firma INDEX BY PLS_INTEGER;
219 |      af_t_ang_firma;
220 |
221 |      -- variabile definite
222 |      v_id_popas_popular      popas.id_popas%TYPE;
223 |      v_locatie_popas_popular locatie%rowtype;
224 |      v_nr_afisari           NUMBER;
225 |
226 |      -- cursor pentru a popula tabelul indexat
227 |      CURSOR c_ang_firma IS SELECT id_angajat, id_firma
228 |                           FROM angajat
229 |                           WHERE tip_angajat = 'PAZNIC';
230 |      BEGIN
231 |          OPEN c_ang_firma; -- deschide cursorul
232 |          FETCH c_ang_firma BULK COLLECT INTO af;
233 |          IF c_ang_firma%rowcount = 0 THEN -- in cazul in care nu s-au obtinut informatii
234 |              RAISE_APPLICATION_ERROR(-20147, 'Nu au putut fi obtinute id-urile paznicilor cu
235 |              \fimile in care lucreaza!');
236 |          END IF;
237 |          CLOSE c_ang_firma; -- inchide cursorul
238 |
239 |          -- extindem tabelele cu informatii despre angajat si depozite folosind
240 |          -- marimea tabelului indexat - extindem acum tabelele pentru a nu o face de
241 |          -- fiecare data cand vrem sa inseram ceva
242 |          ta.EXTEND(af.COUNT);
243 |          td.EXTEND(af.COUNT);
244 |          FOR i IN af.FIRST..af.LAST LOOP
245 |              -- TODO pentru a simula un caz de exceptie inlocuieste mai jos cu valori random
246 |              ta(i) := f14_paznic_info(af(i).id_angajat, af(i).id_firma);
247 |              td(i) := f14_depozit_info(ta(i).id_echipa_paza);
248 |          END LOOP;
249 |          obtine_cel_mai_vizitat_popas(v_id_popas_popular);
250 |          v_locatie_popas_popular := locatie_popas(v_id_popas_popular);
251 |
252 |          -- afisare filtrata a informatiilor din tabele
253 |          v_nr_afisari := 0; -- verificam daca au fost afisate informatii
254 |          FOR i IN af.FIRST..af.LAST LOOP -- ne folosim tot de tabelul indexat
255 |              -- daca inlocuim cu = putem exemplifica exceptia cand nu se realizeaza afisari
256 |              IF td(i).id_locatie <> v_locatie_popas_popular.id_locatie THEN
257 |
258 |                  DBMS_OUTPUT.PUT_LINE('-----');
259 |                  DBMS_OUTPUT.PUT_LINE('Paznicul'||ta(i).nume_angajat||' '||ta(i).
260 |                  \frenume_angajat
261 |                  ||' are salariul de '||ta(i).salariu_actual||' de lei si ar dori ' ||
262 |                  ta(i).salariu_dorit||' de lei.');
263 |                  DBMS_OUTPUT.PUT_LINE('Considerand salariile paznicilor din firma este '
264 |                  ||ta(i).este_platit_bine);
265 |                  DBMS_OUTPUT.PUT_LINE('A lucrat pentru firma '||ta(i).nume_firma
266 |                  ||' timp de '||ta(i).vechime_angajat);
267 |                  -- in functie de nr de colegi vom face o afisare speciala
268 |                  DBMS_OUTPUT.PUT('In momentul de fata este repartizat ');
269 |                  IF td(i).nr_colegi = 0 THEN
270 |                      DBMS_OUTPUT.PUT('singur');
271 |                  ELSIF td(i).nr_colegi = 1 THEN
272 |                      DBMS_OUTPUT.PUT('alatauri de inca un paznic');
273 |                  ELSE
274 |                      DBMS_OUTPUT.PUT('alaturi de inca '||td(i).nr_colegi||' paznici');
275 |                  END IF;
276 |
277 |                  DBMS_OUTPUT.PUT(' intr-un depozit din judetul '||td(i).judet||' localitatea '
278 |                  ||td(i).localitate||' pe strada '||td(i).strada||' la nr.'||td(i).nr);
279 |                  DBMS_OUTPUT.NEW_LINE();
280 |                  v_nr_afisari := v_nr_afisari + 1; -- marcam faptul ca am mai facut o afisare
281 |              END IF;
281 |          END LOOP;

```

```

282 |
283 |     IF v_nr_afisari = 0 THEN -- nu au fost nimic afisat, aruncam o exceptie
284 |         RAISE_APPLICATION_ERROR(-20148, 'Nu au fost gasiti paznici care sa respecte
285 |         → criteriile date');
286 |
287 |     EXCEPTION -- tratarea de exceptii
288 |         -- pentru orice exceptie primita afiseaza mesajul de eroare, deoarece am
289 |         -- creat noi exceptii specializate si detalitate despre ce exceptii ar putea
290 |         -- aparea, in cazul in care alte exceptii neprevazute apar tot va fi afisat
291 |         -- mesajul exceptiei
292 |     WHEN OTHERS THEN
293 |         DBMS_OUTPUT.PUT_LINE(SQLERRM);
294 |     END;
295 |
296 | PROCEDURE p14_info_cer2 IS
297 |     TYPE rec_ang_viz IS RECORD (
298 |         data_sosire      istoric_popasuri.data_sosire%TYPE,
299 |         data_plecare    istoric_popasuri.data_plecare%TYPE,
300 |         nume_ang        angajat.nume%TYPE,
301 |         prenume_ang     angajat.prenume%TYPE,
302 |         nr_telefon_ang angajat.nr_telefon%TYPE,
303 |         nume_firma      firma.nume%TYPE
304 |     );
305 |     TYPE t_ang_viz IS TABLE OF rec_ang_viz;
306 |     t t_ang_viz;
307 |
308 |     v_id_popas popas.id_popas%TYPE;
309 |
310 |     CURSOR c_ist_pop(p_id_popas popas.id_popas%TYPE) IS
311 |         SELECT ip.data_sosire, ip.data_plecare, a.nume, a.prenume, a.nr_telefon, f.nume
312 |             FROM istoric_popasuri ip JOIN angajat a ON (ip.id_angajat = a.id_angajat)
313 |                 JOIN firma f ON (a.id_firma = f.id_firma)
314 |             WHERE ip.id_popas = p_id_popas
315 |                 ORDER BY ip.data_sosire desc; -- sorteaza descrescator in functie de cand au
316 |             → venit
317 |     BEGIN
318 |         -- pentru cel mai vizitat popas afiseaza informatii despre soferii care au
319 |         -- venit la el
320 |         obtine_cel_mai_vizitat_popas(v_id_popas);
321 |
322 |         OPEN c_ist_pop(v_id_popas);
323 |         FETCH c_ist_pop BULK COLLECT INTO t;
324 |         IF c_ist_pop%rowcount = 0 THEN -- pune 1 pentru a testa exceptia
325 |             RAISE_APPLICATION_ERROR(-20149, 'Nu a vizitat niciun sofer motelul cu id-ul ' ||
326 |             → v_id_popas);
327 |         END IF;
328 |         CLOSE c_ist_pop;
329 |         FOR i IN t.FIRST..t.LAST LOOP
330 |             DBMS_OUTPUT.PUT_LINE('La data de '||t(i).data_sosire||' a venit soferul '
331 |                 ||t(i).nume_ang||' '||t(i).prenume_ang||' cu nr. de telefon '|
332 |                 t(i).nr_telefon_ang||' angajat la'||t(i).nume_firma||' si a plecat la '|
333 |                 ||t(i).data_plecare);
334 |         END LOOP;
335 |         DBMS_OUTPUT.PUT_LINE('-----');
336 |         DBMS_OUTPUT.PUT_LINE('-----');
337 |         DBMS_OUTPUT.NEW_LINE(); -- separator intre cerinte
338 |         -- afiseaza aceleasi informatii pentru toate motelele
339 |         -- ciclu cursor cu subcerere
340 |         FOR p IN (SELECT id_popas FROM popas WHERE tip_popas = 'MOTEL') LOOP
341 |             -- sterge informatiile din tabel pentru a retine alte date
342 |             t.DELETE();
343 |             -- obtine un istoric al opririlor soferilor la motelul curent
344 |             OPEN c_ist_pop(p.id_popas);
345 |             FETCH c_ist_pop BULK COLLECT INTO t LIMIT 3; -- obtine doar primele 3
346 |             IF c_ist_pop%rowcount = 0 THEN -- pune 1 pentru a testa exceptia
347 |                 RAISE_APPLICATION_ERROR(-20149, 'Nu a vizitat niciun sofer motelul cu id-ul
348 |                 → '||p.id_popas);
349 |             END IF;
350 |             CLOSE c_ist_pop;
351 |
352 |             FOR i IN t.FIRST..t.LAST LOOP

```

```

351 |           DBMS_OUTPUT.PUT_LINE('La data de'||t(i).data_sosire||' a venit soferul '
352 |           ||t(i).nume_ang||' '||t(i).prenume_ang||' cu nr. de telefon '||
353 |           t(i).nr_telefon_ang||' angajat la'||t(i).nume_firma||' si a plecat la '|
354 |           ||t(i).data_plecare);
355 |
356 |       END LOOP;
357 |   END LOOP;
358 | EXCEPTION -- tratarea de exceptii
359 |   -- pentru orice exceptie primita afiseaza mesajul de eroare, deoarece am
360 |   -- creat noi exceptii specializate si detalitate despre ce exceptii ar putea
361 |   -- aparea, in cazul in care alte exceptii neprevazute apar tot va fi afisat
362 |   -- mesajul exceptiei
363 |   WHEN OTHERS THEN
364 |       DBMS_OUTPUT.PUT_LINE(SQLERRM);
365 |   END;
366 |
367 | END;
368 |
369 |
370 | EXECUTE pack_ex14.p14_info_cer1;
371 | EXECUTE pack_ex14.p14_info_cer2;

```

6 Anexe: Dovezi rulare corecta a codului

Oracle SQL Developer

Connections

project : ISTORIC_CAMIOANE_CONDUSE

Columns Data Model Constraints Grants Statistics Triggers Dependencies Details Partitions Indexes SQL

COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1 ID ISTORIC_CAMIOANE_CONDUSE	NUMBER(10,0)	No	(null)	1	(null)
2 ID ANGAJAT	NUMBER(10,0)	Yes	(null)	2	(null)
3 ID CAMION	NUMBER(5,0)	Yes	(null)	3	(null)
4 DATA_INCEPUT	DATE	Yes	sysdate	4	(null)
5 DATA_SFARSIT	DATE	Yes	(null)	5	(null)

Tabele create

Oracle SQL Developer : project

Connections

project

Worksheet Query Builder

```
1 SELECT * FROM angajat;
```

Query Result

ID_ANGAJAT	NUME	PRENUME	NR_TELEFON	DATA_ANGAJARE	SALARU	ID_FIRMA	TIPO_ANGAJAT	ID_ECHIPA_PAZA
1	20000	Ion	Vasile	0722123456	11-APR-20	5000	1100 SOFER	(null)
2	20020	Ovidiu	Ion	0723123456	12-MAY-20	4000	1100 SOFER	(null)
3	20040	Popescu	Andrei	0723323456	11-MAR-20	6000	1200 SOFER	(null)
4	20060	Marian	Gheorghe	0721213456	11-NOV-20	3000	1200 SOFER	(null)
5	20080	Manea	George	0721123359	03-MAR-20	4000	1300 SOFER	(null)
6	20100	Constantin	Gheorghe	0735123456	09-APR-20	4500	1400 SOFER	(null)
7	20120	Ion	Catalin	0721126566	02-JUL-20	3400	1500 SOFER	(null)
8	20140	Marius	Catalin	072119356	02-JUL-20	3300	600 PAZNIC	1
9	20160	Grant	Ion	0723123456	02-JUL-20	3175	600 PAZNIC	1
10	20180	Ivan	George	0726100456	02-JUL-20	3250	600 PAZNIC	2
11	20200	Neagu	Ion	0728123456	02-JUL-20	3575	800 PAZNIC	3
12	20220	Popescu	Cristian	0724523456	02-JUL-20	3600	700 PAZNIC	4
13	20240	Marius	Mihai	07212233456	02-JUL-20	3525	900 PAZNIC	5
14	20260	Mihai	Gheorghe	0724323456	02-JUL-20	3375	1000 PAZNIC	6
15	20280	Minin	Catalin	0721123456	02-JUL-20	3225	1000 PAZNIC	7
16	20300	Andrei	David	0722123456	02-JUL-20	3250	600 PAZNIC	8
17	20320	Popescu	Stefan	0721145456	02-JUL-20	3475	700 PAZNIC	9
18	20340	Ion	Catalin	07211765456	02-JUL-20	3320	600 PAZNIC	10

Date salvate in tabelul ANGAJAT

Oracle SQL Developer : project

File Edit View Navigate Run Source Team Tools Window Help

Connections Oracle Connections BD BD_USER project

Tables ANGAJAT CAMPANIE DEPOZIT ECHIPAMENTI FIRMA INVENTAR_DEPOZIT INVENTAR_TRANSPORT ISTORIC_CAMPANIE_CDUSUE ISTORIC_POPASURI LOT_MARFA PERMS POPAS TRANSPORT

Views Indexes Packages Functions Operators Triggers Sequences Materialized Views Public Synonyms Database Links Directories Editions Application Express XML Schemas XML DB Repository Scheduler Recycle Bin Other Users Database Schema Service Connections

Worksheet Query Builder

```
1 SELECT * FROM transport;
```

Query Result SQL All Rows Fetched: 11 in 0.003 seconds

	ID_TRANSPORT	DEPOZIT_PLECARE	DEPOZIT_DESTINATIE	ID_CAMPON	DATA_PLECARE
1	10000	3000	500	15.06-MAY-21	
2	10010	3000	500	25.08-MAY-21	
3	10020	3500	1000	35.06-MAY-21	
4	10030	4000	1000	45.06-MAY-21	
5	10040	4500	1000	55.06-MAY-21	
6	10050	5000	1500	11.07-MAY-21	
7	10060	5000	1500	105.05-MAY-21	
8	10070	3500	2000	85.06-MAY-21	
9	10080	3500	2000	85.07-MAY-21	
10	10090	3000	2500	25.10-MAY-21	
11	10100	3000	2500	15.11-MAY-21	

| Line 1 Column 24 | Insert | Modified | Windows: CR

Date salvate in tabelul TRANSPORT

Oracle SQL Developer : project

File Edit View Navigate Run Source Team Tools Window Help

Connections Oracle Connections BD BD_USER project

Tables ANGAJAT CAMPANIE DEPOZIT ECHIPAMENTI FIRMA INVENTAR_DEPOZIT INVENTAR_TRANSPORT ISTORIC_CAMPANIE_CDUSUE ISTORIC_POPASURI LOT_MARFA PERMS POPAS TRANSPORT

Views Indexes Packages Functions Operators Triggers Sequences Materialized Views Public Synonyms Database Links Directories Editions Application Express XML Schemas XML DB Repository Scheduler Recycle Bin Other Users Database Schema Service Connections

Worksheet Query Builder

```
1 SELECT * FROM istoric_popasuri;
```

Query Result SQL All Rows Fetched: 10 in 0.002 seconds

	ID_ISTORIC_POPAS	ID_POPAS	ID_ANGAJAT	DATA_SOSIRE	DATA_PLECARE
1	50	180	20000	06-MAY-21	06-MAY-21
2	55	120	20000	06-MAY-21	06-MAY-21
3	60	180	20020	08-MAY-21	08-MAY-21
4	65	180	20020	08-MAY-21	08-MAY-21
5	70	100	20000	06-MAY-21	06-MAY-21
6	75	140	20060	06-MAY-21	06-MAY-21
7	80	140	20080	06-MAY-21	06-MAY-21
8	85	140	20100	06-MAY-21	06-MAY-21
9	90	180	20120	07-MAY-21	07-MAY-21
10	95	180	20120	05-MAY-21	05-MAY-21

| Line 1 Column 31 | Insert | Modified | Windows: CR

Detailed description of the screenshot: The image shows two Oracle SQL Developer windows side-by-side. Both windows have tabs for 'project' and 'script'. The left window displays a PL/SQL procedure named 'TRACKING_PROC'. It includes several DBMS_OUTPUT.PUT_LINE statements that print status messages and transport details to the right window's output pane. The right window is titled 'Script Output' and shows the actual output of the procedure's execution. The output contains multiple entries for different trucks (VOLVO, SCANIA, DAF) with their respective license plates, company names (Road Logistics, Transbiso, FedEx), and transport details like date, driver name, and payment status. The output is timestamped with '06:00:00' and includes line numbers corresponding to the code in the left window.

```

50  TYPE rec_transport IS RECORD (
51    nume_softer      angajat.nume_type,
52    prenume_softer   angajat.prenume_type,
53    salarie          angajat.salariu_type,
54    data_transport   varchar2(255)
55  );
56  END;
57  BEGIN
58    OPEN c_camion; -- deschide cursorul principal
59    LOOP
60      FETCH c_camion INTO v_id_camion, v_marca_camion, v_nr_immatriculare_camion,
61      v_nume_determinant_camion, v_cursur_transporturi;
62      EXIT WHEN c_camion%NOTFOUND; -- nu mai sunt camioane de afisate
63      DBMS_OUTPUT.PUT_LINE('-----');
64      -- afiseaza informatii despre camion
65      DBMS_OUTPUT.PUT_LINE('Camionul cu marca ''||v_marca_camion||'' avand nr. de immatriculare
66      ''||v_nr_immatriculare_camion||'' si este detinut de compania '
67      ||v_nume_determinant_camion||',');
68      DBMS_OUTPUT.PUT_LINE('status transporturi:');
69      -- pentru fiecare camion afisam informatiile despre status transporturi
70      LOOP
71        PETCH v_cursur_transporturi INTO cur;
72        -- lezi din loop atunci cand nu mai sunt transporturi
73        EXIT WHEN v_cursur_transporturi%NOTFOUND;
74        DBMS_OUTPUT.PUT_LINE('-----');
75        -- afiseaza informatii despre transport si sofer
76        DBMS_OUTPUT.PUT_LINE('Transport in data de ''||cur.data_transport||'' realizat de '
77        ||cur.nume_sofer||' ''||cur.prenume_sofer||'' care este platit lunar cu '
78        ||cur.salariu||' lei.');
79      END LOOP;
80    END LOOP;
81    DBMS_OUTPUT.PUT_LINE('-----'); -- separam cu o linie la final
82  END;
83  CLOSE c_camion; -- inchidem cursorul
84;
85  -- executa
86  BEGIN
87    EXECUTE IMMEDIATE 'DROP PROCEDURE tracking_proc';
88  EXCEPTION
89  WHEN OTHERS THEN
90    IF SQLCODE <> -4044 THEN
91      DBMS_OUTPUT.PUT_LINE('Nu exista procedura tracking_proc');
92    END IF;
93  END;
94;
95  Procedure TRACKING_PROC compiled
96;
97  PL/SQL procedure successfully completed.

```



```

54  (SELECT count(id_camion) -- numarul cate camioane a condus soferul folosind o subcerere
55  FROM istoric_camoane_conduse sic
56  WHERE sic.id_angajat = a.id_angajat) nr_camoane_conduse
57  INTO nr_camoane_conduse;
58  FROM angajat A JOIN istoric_camoane_conduse ioc ON (a.id_angajat = ioc.id_angajat)
59  JOIN camion C ON (ioc.id_camion = c.id_camion)
60  JOIN firma f ON (a.id_firma = f.id_firma)
61  WHERE ioc.data_eferat IS NULL -- null marcheaza faptul ca incă conduce camionul
62  AND a.nume = p_nume AND a.prenume = p_prenume;
63  RETURN resultat;
64  END;
65  /
66  /DECLARE
67  /  v_ang tab_ang_info%TYPE;
68  BEGIN
69    v_ang := func_ang_info_sofer('Ion', 'Vasile'); -- ex bun
70    DBMS_OUTPUT.PUT_LINE('Soferul posteaza fi contextul la nr.'||v_ang.telefon_ang);
71    -- v_ang := func_ang_info_sofer('Andrei', 'Vasile'); -- no data found
72    -- v_ang := func_ang_info_sofer('Ion', 'Catalin'); -- too many rows
73    -- v_ang := func_ang_info_sofer('Andrei', 'David'); -- exception specifica nu este sofer
74    DBMS_OUTPUT.PUT_LINE('-----');
75    -- o simpla afisare a datelor pentru a verifica caul bun
76    DBMS_OUTPUT.PUT_LINE('Soferul posteaza fi contextul la nr.'||v_ang.telefon_ang);
77    DBMS_OUTPUT.PUT_LINE('Salariul este de '||v_ang.salariu_ang'||'. Conduse in total ''||v_ang.nr_camoane_conduse
78    ||'' pana acum. In momentul de fata conduce un camion ''||v_ang.marca_camion
79    ||'' cu nr. de immatriculare ''||v_ang.nr_immatriculare||''.');
80  EXCEPTION
81    WHEN NO_DATA_FOUND THEN
82      DBMS_OUTPUT.PUT_LINE('Nu au putut fi obtinute informatii despre soferul dat');
83    WHEN TOO_MANY_ROWS THEN
84      DBMS_OUTPUT.PUT_LINE('Nu au putut fi obtinute informatii despre soferul dat');
85    WHEN p8_exceptions.no_eta_sofer THEN
86      DBMS_OUTPUT.PUT_LINE('Angajatul nu este sofer. Nu putem intotdeauna informatii despre el.');
87    WHEN OTHERS THEN
88      DBMS_OUTPUT.PUT_LINE('Cod eroare: '||SQLCODE);
89      DBMS_OUTPUT.PUT_LINE('Mesaj eroare: '||SQLERRM);
90  END;
91  /

```

Script Output: Task completed in 0.072 seconds

Table TAB8_ANG_INFO created.

Package PR_EXCEPTIONS compiled

Function FUNC8_INFO_SOFER compiled

Soferul posteaza fi contact la nr 0722123456. Lucreaza pentru firma Road Logistics unde primeste salariul de \$000. A condus in total 3 pana acum. In momentul de fata conduce un camion VOLVO cu nr. de immatriculare CT12FSD

PL/SQL procedure successfully completed.

The screenshot shows the Oracle SQL Developer interface with a running PL/SQL script. The code handles transport requests, including validating truck numbers and updating transport status. It includes comments explaining the logic for handling different transport types (THIN, OTHERS) and specific transport IDs (10100, 101000). The script also manages updates to the 'camion' and 'transport' tables, including a rollback section and a final commit. The execution log at the bottom shows the script was completed in 0.04 seconds with 1 row updated.

```
1 WHEN transport_fara_marfa THENIN -- daca am gasit transportul dar nu si marfa
2   DBMS_OUTPUT.PUT_LINE('Camionul ''||p_nr_immatriculare||'
3                         ||' nu are marfa in transportul cu id=||p_id_transport||');
4   ' dar nu exista inregistrari pentru marfa transportata. Transport invalid.');
5 WHEN OTHERS THEN -- alte exceptiee neprevazute
6   DBMS_OUTPUT.PUT_LINE(SQLERRM);
7 END;
8
9
10-- 10100 id transport pentru care s-a stergut marfa transportata pentru a fi tot o eroare
11-- CULCED nr immatriculare data plecare 11-05-2021
12 select * from transport WHERE id_transport = 10100;
13 select * from camion where id_camion =15;
14 update transport set marfa_inventar=0 where id_transport = 10100;
15 DELETE FROM inventar_transport WHERE id_transport = 10100;
16
179-- actualizam transportul cu id-ul 10000 pentru data de 10000
18-- pentru ca sunt doar 2 transporturi la acelasi data dar nu am gasit intrusat acelasi camion
19-- nu putem rezolva 2 transporturi in acei momenti de la acelasi timp
20
21 UPDATE transport
22 SET data_plecare = to_date('10-05-2021', 'dd-mm-yyyy')
23 WHERE id_transport = 10000;
24
25 select * from transport where id.camion =25;
26 select * from camion where id.camion =25;
27
28 rollback; -- anulam modificarile facute
29
30 execute procedure
31
32 BEGIN
33   proc_afisare_transport('MM57RXF', '05-05-2021'); -- ex bun
34   proc_afisare_transport('MM57RXF', '10-05-2021'); -- no data found - nu am gasit transportul
35   -- proc9_afisare_transport('CT930BMM', '10-05-2021'); -- too many rows - am gasit mai multe transporturi
36   proc_afisare_transport('C911FED', '11-05-2021'); -- custom error - am gasit transportul dar nu si marfa
37 END;
38
39
40 Log Output: *
41 Task completed in 0.04 seconds
42 U ROWS Updated:
43
44
45 1 row updated.
46
47 Camionul MM57RXF immatriculat in judetul Maramures a realizat un transport acum 8 in data de 05-05-2021 06:00
48 Camionul a plecat din depositul detinut de FedEx din Loc: Bucuresti Str. 1 mai Nr. 1213 spre depositul detinut de ABOUT YOU din Loc: Constanta Str. Victoriei Nr. 123
49 Marfa transportata este de camion:
50 numarul-traseu=100
51 imbracamente-hanorace=10
52
53
54 PL/SQL procedure successfully completed.
55
56 Camionul MM57RXF nu a realizat niciun transport in data de 10-05-2021
57
58 PL/SQL procedure successfully completed.
59
60
61 Compiler Log Click on a identifier with the Control key down to perform 'Go To Definition' | 100% | Help | About
```

The screenshot shows the Oracle SQL Developer interface with a PL/SQL script editor and a results tab.

PL/SQL Script Content:

```
1  Oracle SQL Developer - project
2  Edit  View  Favorites  Bookmarks  Source  Tools  Help
3
4  Window Home  Database  Project  Help  About
5  Home  Recent  Database  Project  Help  About
6  Worksheet  Query Builder
7
8  WHEN transport_fara_marfa THEN -- daca am gasit transportul dar nu si marfa
9  DBMS_OUTPUT.PUT_LINE('Camionul ''||p_nr_immatriculare||'
10  ||' nu are marfa transportata.'||p_data_transport);
11  ' dar nu exista inregistrari pentru marfa transportata. Transport invalid.');
12  WHEN OTHERS THEN -- alte exceptiee neprevazute
13  DBMS_OUTPUT.PUT_LINE(SQLERRM);
14  END;
15
16
17  -- 10100 id transport pentru a sterge marfa transportata pentru a forta o eroare
18  -- CUILEDID nr immatriculare data plecare 11-05-2021
19  select * from transport WHERE id_transport = 10100;
20  select * from camion where id_camion =15;
21  select * from inventar_transport where id_inventar=10100;
22  DELETE FROM inventar_transport WHERE id_transport = 10100;
23
24  /* actualizam transportul cu id-ul 10090 pentru data de 10090
25  pentru a face ca transportul sa devină invalidat nu avand intrare acelasi camion
26  nu putem rezilii 2 transporturi in aceiasi zi din cauza limitelor in acelasi timp */
27  UPDATE transport
28  SET data_plecare = to_date('10-05-2021', 'dd-mm-yyyy')
29  WHERE id_transport = 10090;
30
31  select * from transport where id.camion =25;
32  select * from camion where id.camion =25;
33
34  rollback; -- anulam modificarile facute
35
36  execute procedure
37
38  BEGIN
39      proc_afisare_transport('MM57RXF', '05-05-2021'); -- ex bun
40      -- proc_afisare_transport('MM57RXF', '10-05-2021'); -- no data found - nu am gasit transportul
41      -- proc_afisare_transport('CT940MM', '10-05-2021'); -- too many rows - am gasit mai multe transporturi
42      -- proc_afisare_transport('CL12PFD', '11-05-2021'); -- custom error - am gasit transportul dar nu si marfa
43  END;
44
```

Log Output:

- PL/SQL procedure successfully completed.
- Camionul MM57RXF nu a realizat niciun transport in data de 10-05-2021
- PL/SQL procedure successfully completed.
- Camionul CT940MM a realizat mai multe transporturi in data de 10-05-2021. Date eronate.
- PL/SQL procedure successfully completed.
- Camionul CL12PFD a realizat un transport in data de 11-05-2021 dar nu exista inregistrari pentru marfa transportata. Transport invalid.
- PL/SQL procedure successfully completed.

Bottom Status Bar:

Console Log Click on a file to perform 'Go to Definition' | 100% | Help | Recent

```
Oracle SQL Developer project

File Edit View Navigator Run Source Tools Window Help

Connections
+ - [+] project
  + - [+] Create Connections
    + - [+] BD_USER
    + - [+] BD_DEVELOPER
    + - [+] project
      + - [+] Tables
        + - [+] ANGAJAT
          + - [+] Insert
            + - [+] 45: INSERT INTO angajat
            + - [+] 46: VALUES (SEQ_ANG.NEXTVAL, 'Ion', 'Vasile', '0722123456',
            + - [+] 47: to_date('11-04-2020 12:00', 'dd-mm-yyyy hh24:mi'), 5000, 1100, 'SOFER', null);
            + - [+] 48: 
            + - [+] 49: DELETE FROM angajat
            + - [+] 50: WHERE id_angajat = 20000;
            + - [+] 51: 
            + - [+] 52: UPDATE angajat
            + - [+] 53: SET salaryin = 10000
            + - [+] 54: WHERE id_angajat = 20000;
            + - [+] 55: 
            + - [+] 56: UPDATE ANGAJAT
            + - [+] 57: SET salaryin = 10000
            + - [+] 58: WHERE id_angajat = 20000;
            + - [+] 59: 
            + - [+] 60: Trigger TRIG_CHECK_ANG compiled

Trigger TRIG_CHECK_ANG compiled

Error starting at line : 45 in command -
INSERT INTO angajat
VALUES (SEQ_ANG.NEXTVAL, 'Ion', 'Vasile', '0722123456',
to_date('11-04-2020 12:00', 'dd-mm-yyyy hh24:mi'), 5000, 1100, 'SOFER', null)
Error starting at line : 50 in command -
DELETE FROM angajat
WHERE id_angajat = 20000
ORA-20111: Inserarea din tabelul de angajati este permisa doar in timpul programului de lucru!
ORA-06512: at "PROJECT.TRIG_CHECK_ANG", line 9
ORA-04088: error during execution of trigger 'PROJECT.TRIG_CHECK_ANG'

Trigger TRIG_CHECK_ANG compiled

Trigger TRIG_CHECK_ANG compiled

Error starting at line : 49 in command -
DELETE FROM angajat
WHERE id_angajat = 20000
Error starting at line : 50 in command -
UPDATE angajat
SET salaryin = 10000
WHERE id_angajat = 20000
Error starting at line : 51 in command -
UPDATE ANGAJAT
SET salaryin = 10000
WHERE id_angajat = 20000
Error starting at line : 52 in command -
UPDATE ANGAJAT
SET salaryin = 10000
WHERE id_angajat = 20000
ORA-20111: Nu se poate schimba id-ul unui angajat!
ORA-06512: at "PROJECT.TRIG_CHECK_ANG", line 12
ORA-04088: error during execution of trigger 'PROJECT.TRIG_CHECK_ANG'

Trigger TRIG_CHECK_ANG compiled

Trigger TRIG_CHECK_ANG compiled

Error starting at line : 56 in command -
UPDATE ANGAJAT
SET id_angajat = 40000
WHERE id_angajat = 20000
Error starting at line : 57 in command -
UPDATE ANGAJAT
SET id_angajat = 40000
WHERE id_angajat = 20000
ORA-20114: Nu se poate schimba id-ul unui angajat!
ORA-06512: at "PROJECT.TRIG_CHECK_ANG", line 23
ORA-04088: error during execution of trigger 'PROJECT.TRIG_CHECK_ANG'

Trigger TRIG_CHECK_ANG compiled

Compiler Log Click on an identifier with the Control key down to perform 'Go to Declaration' | Line 23 Column 36 | Insert | Help/Help Contents
```

The screenshot shows the Oracle SQL Developer interface. The left sidebar displays a tree view of database connections, schemas, and objects. The main workspace contains several tabs:

- Script**: Displays a PL/SQL script with numerous comments and code snippets related to inserting data into the ANGAJAT table.
- Output**: Shows the results of the executed script, including error messages for trigger compilation and execution.
- Errors**: Shows the errors from the execution of the TRIG_CHECK_ANG trigger.

The bottom status bar indicates "Compiler Log" and "Click on an identifier with the Control key down to perform 'Go to Declaration'".

```
/* Oracle SQL Developer project */
-- File Edit View Source Tools Window Help
-- project
-- 3D ENDY
-- 39
-- 40 -- coment corect a testa cand in cale azi ar fi tii din weekend
-- 41 select to_char(sysdate,'09-01-2022','dd-mm-yyyy'), 'DVI'
-- from dual;
-- 42
-- 43
-- 44
-- 45 INSERT INTO angajat
-- 46 VALUES (SEQ_ANG_NEXKVAL, 'Ion', 'Vasile', '0722123456',
-- 47 to_date('11-04-2020 12:00', 'dd-mm-yyyy hh24:mi'), 5000, 1100, 'SOFER', null);
-- 48 DELETE FROM angajat
-- 49 WHERE id_angajat = 20000
-- 50
-- 51 UPDATE angajat
-- 52 SET salariei = 10000
-- 53 WHERE id_angajat = 20000
-- 54
-- 55 UPDATE angajat
-- 56 SET tip_angajat = 'PANZIC'
-- 57 WHERE id_angajat = 40000
-- 58 WHERE id_angajat = 20000
-- 59
-- 60 UPDATE ANGAJAT
-- 61 SET tip_angajat = 'PANZIC'
-- 62 WHERE id_angajat = 20000
-- 63
-- 64 -- coment corect in functie de trigger
-- 65 INSERT INTO angajat
-- 66 VALUES (SEQ_ANG_NEXKVAL, 'Ion', 'Vasile', '0722123456',
-- 67 to_date('11-04-2020 12:00', 'dd-mm-yyyy hh24:mi'), 5000, 1100, 'SOFER', null);

Trigger TRIG_CHECK_ANG compiled

Error starting at line : 56 in command -
UPDATE ANGAJAT
SET id_angajat = 40000
WHERE id_angajat = 20000
Error report
ORA-06512: at ""PROJECT.TRIG_CHECK_ANG", line 23
ORA-04008: error during execution of trigger 'PROJECT.TRIG_CHECK_ANG'

Error starting at line : 60 in command -
UPDATE ANGAJAT
SET tip_angajat = 'PANZIC'
WHERE id_angajat = 20000
Error report
ORA-20114: Nu se poate schimba tipul unui angajat!
ORA-06512: at ""PROJECT.TRIG_CHECK_ANG", line 30
ORA-04008: error during execution of trigger 'PROJECT.TRIG_CHECK_ANG'
```

Script Output x Query Result x Query Result 1 x Query Result 2

SQL | All Rows Fetched: 15 in 0.004 seconds

ID_FIRMA	NUME	TIP_FIRMA
1	100 Lidl	CLIENT
2	200 Profi	CLIENT
3	300 ABOUT YOU	CLIENT
4	400 Deichmann	CLIENT
5	500 H and M	CLIENT
6	600 STOP SEAL GUARD	PAZA
7	700 ATMAN PROTECTION	PAZA
8	800 GARDIENII	PAZA
9	900 QUICK PROTECT	PAZA
10	1000 JOHNNY VIP SECURITY	PAZA
11	1100 Road Logistics	TRANSPORT
12	1200 Transibo	TRANSPORT
13	1300 EasyCargo	TRANSPORT
14	1400 Lextom Trans Asd	TRANSPORT
15	1500 FedEx	TRANSPORT

Script Output | Query Result | Query Result 1 | Query Result

SQL | All Rows Fetched: 10 in 0.001 seconds

	ID_DEPOZIT	ID_LOCATIE	ID_ECHIPA_PAZA	ID_FIRMA	
1	500	30	2	100	
2	1000	50	3	200	
3	1500	80	1	300	
4	2000	90	5	400	
5	2500	110	6	500	
6	3000	100	4	1100	
7	3500	120	10	1200	
8	4000	140	8	1300	
9	4500	150	7	1400	
10	5000	130	9	1500	

Script Output | Query Result | Query Result 1 | Query Result 2

SQL | All Rows Fetched: 11 in 0.003 seconds

	ID_CAMION	MARCA	NR_INMATRICULARE	ID_FIRMA	
1	15	VOLVO	CT12FSD	1100	
2	25	VOLVO	CT90MMM	1100	
3	35	SCANIA	PH13SCN	1200	
4	45	DAF	B55ASD	1300	
5	55	IVECO	MM13IVC	1400	
6	65	MAN	TM20QWE	1100	
7	75	SCANIA	IS32ISA	1100	
8	85	MAN	IS01MAN	1200	
9	95	DAF	TM11IOP	1300	
10	105	VOLVO	MM57RXF	1500	
11	115	SCANIA	B29SCN	1500	

Script Output | Query Result | Query Result 1 | Query Result 2 | Query Result 3 | Query Result 4 | All Rows Fetched: 18 in 0.003 seconds

	ID_ANGAJAT	NUME	PRENUME	NR_TELEFON	DATA_ANGAJARE	SALARIU	ID_FIRMA	TIP_ANGAJAT	ID_ECHIPA_PAZA
1	20000	Ion	Vasile	0722123456	11-APR-20	5000	1100	SOFER	(null)
2	20020	Ovidiu	Ion	0727124356	12-MAY-20	4000	1100	SOFER	(null)
3	20040	Popescu	Andrei	0723323465	11-MAR-20	6000	1200	SOFER	(null)
4	20060	Marian	Gheorghe	0721213456	11-NOV-20	3000	1200	SOFER	(null)
5	20080	Manea	George	0721123959	03-MAR-20	4000	1300	SOFER	(null)
6	20100	Constantin	Gheorghe	0735123456	09-APR-20	4500	1400	SOFER	(null)
7	20120	Ion	Catalin	0721126566	02-JUL-20	3400	1500	SOFER	(null)
8	20140	Marius	Catalin	072119356	02-JUL-20	3300	600	PAZNIC	1
9	20160	Grant	Ion	0723123456	02-JUL-20	3100	600	PAZNIC	1
10	20180	Ivan	George	07226100456	02-JUL-20	3250	600	PAZNIC	2
11	20200	Neagu	Ion	0728123456	02-JUL-20	3500	800	PAZNIC	3
12	20220	Popescu	Cristian	0724523456	02-JUL-20	3600	700	PAZNIC	4
13	20240	Marius	Mihai	0721223456	02-JUL-20	3450	900	PAZNIC	5
14	20260	Radu	Gheorghe	0724323456	02-JUL-20	3375	1000	PAZNIC	6
15	20280	Marin	Catalin	0725623456	02-JUL-20	3150	1000	PAZNIC	7
16	20300	Andrei	David	0722123456	02-JUL-20	3250	600	PAZNIC	8
17	20320	Popescu	Stefan	0721145456	02-JUL-20	3400	700	PAZNIC	9
18	20340	Ion	Catalin	0721765456	02-JUL-20	3320	600	PAZNIC	10

Script Output | Query Result | Query Result 1 | All Rows Fetched: 10 in 0.002 seconds

	ID_ECHIPA_PAZA	ID_FIRMA
1	1	600
2	2	600
3	3	800
4	4	700
5	5	900
6	6	1000
7	7	1000
8	8	600
9	9	700
10	10	600

```

D:\Oracle\SQL Developer\project
File Edit View Source Team Tools Window Help
File Recent Project Database Query Editor
Worksheet > QueryEditor
18 UPDATE camion_copy
19   SET id_firma = :NEW.id_firma
20 WHERE id_firma = :OLD.id_firma;
21
22 UPDATE angajat_copy
23   SET id_firma = :NEW.id_firma
24 WHERE id_firma = :OLD.id_firma;
25
26 UPDATE depozit_copy
27   SET id_firma = :NEW.id_firma
28 WHERE id_firma = :OLD.id_firma)
29
30 END;
31
32 -- teste pentru fiecare tip de firma
33 -- 100 600 1100
34 UPDATE firma_copy
35   SET id_firma = 100;
36 WHERE id_firma = 100;
37
38 UPDATE firma_copy
39   SET id_firma = 600;
40 WHERE id_firma = 600;
41
42 UPDATE firma_copy
43   SET id_firma = 1100;
44 WHERE id_firma = 1100;
45
46
47 -- verificare
48 select * from firma_copy;
49 select * from deposit;
50 select * from camion_copy;
51 select * from angajat_copy;
52 select * from depozit_paza_copy;
53
54
Script Output > [Query Result 1] > [Query Result 2] > [Query Result 3] > [Query Result 4]
Table ECHIVA_PAZA_COPY created.

Table FIRMA_COPY created.

Trigger UPDATE_PR_FIRMA compiled

1 row updated.

1 row updated.

1 row updated.

[Compiler Log] Click on an identifier with the Ctrl key down to perform 'Go to Declaration'

```

Script Output X | Query Result X | Query Result 1 X | Query Result 2 X

SQL | All Rows Fetched: 15 in 0.001 seconds

ID_FIRMA	NUME	TIP_FIRMA
1	101 Lidl	CLIENT
2	200 Profi	CLIENT
3	300 ABOUT YOU	CLIENT
4	400 Deichmann	CLIENT
5	500 H and M	CLIENT
6	601 STOP SEAL GUARD	PAZA
7	700 ATMAN PROTECTION	PAZA
8	800 GARDIENII	PAZA
9	900 QUICK PROTECT	PAZA
10	1000 JOHNNY VIP SECURITY	PAZA
11	1101 Road Logistics	TRANSPORT
12	1200 Transibo	TRANSPORT
13	1300 EasyCargo	TRANSPORT
14	1400 Lextom Trans Asd	TRANSPORT
15	1500 FedEx	TRANSPORT

Script Output | Query Result | **Query Result 1** | Query Result 2

SQL | All Rows Fetched: 10 in 0.001 seconds

	ID_DEPOZIT	ID_LOCATIE	ID_ECHIPA_PAZA	ID_FIRMA	
1	500	30	2	100	
2	1000	50	3	200	
3	1500	80	1	300	
4	2000	90	5	400	
5	2500	110	6	500	
6	3000	100	4	1100	
7	3500	120	10	1200	
8	4000	140	8	1300	
9	4500	150	7	1400	
10	5000	130	9	1500	

Script Output | Query Result | **Query Result 1** | Query Result 2

SQL | All Rows Fetched: 11 in 0.001 seconds

	ID_CAMION	MARCA	NR_INMATRICULARE	ID_FIRMA
1	15 VOLVO	CT12FSD		1101
2	25 VOLVO	CT90MMM		1101
3	35 SCANIA	PH13SCN		1200
4	45 DAF	B55ASD		1300
5	55 IVECO	MM13IVC		1400
6	65 MAN	TM20QWE		1101
7	75 SCANIA	IS32ISA		1101
8	85 MAN	IS01MAN		1200
9	95 DAF	TM11IOP		1300
10	105 VOLVO	MM57RXF		1500
11	115 SCANIA	B29SCN		1500

Script Output | Query Result | Query Result 1 | Query Result 2 | Query Result 3 | Query Result 4

SQL | All Rows Fetched: 18 in 0.001 seconds

ID_ANGAJAT	NUME	PRENUME	NR_TELEFON	DATA_ANGAJARE	SALARIU	ID_FIRMA	TIP_ANGAJAT	ID_ECHIPA_PAZA
1	20000	Ion	Vasile	0722123456	11-APR-20	5000	1101 SOFER	(null)
2	20020	Ovidiu	Ion	0727124356	12-MAY-20	4000	1101 SOFER	(null)
3	20040	Popescu	Andrei	0723323465	11-MAR-20	6000	1200 SOFER	(null)
4	20060	Marian	Gheorghe	0721213456	11-NOV-20	3000	1200 SOFER	(null)
5	20080	Manea	George	0721123959	03-MAR-20	4000	1300 SOFER	(null)
6	20100	Constantin	Gheorghe	0735123456	09-APR-20	4500	1400 SOFER	(null)
7	20120	Ion	Catalin	0721126566	02-JUL-20	3400	1500 SOFER	(null)
8	20140	Marius	Catalin	072119356	02-JUL-20	3300	601 PAZNIC	1
9	20160	Grant	Ion	0723123456	02-JUL-20	3100	601 PAZNIC	1
10	20180	Ivan	George	0726100456	02-JUL-20	3250	601 PAZNIC	2
11	20200	Neagu	Ion	0728123456	02-JUL-20	3500	800 PAZNIC	3
12	20220	Popescu	Cristian	0724523456	02-JUL-20	3600	700 PAZNIC	4
13	20240	Marius	Mihai	0721223456	02-JUL-20	3450	900 PAZNIC	5
14	20260	Radu	Gheorghe	0724323456	02-JUL-20	3375	1000 PAZNIC	6
15	20280	Marin	Catalin	0725623456	02-JUL-20	3150	1000 PAZNIC	7
16	20300	Andrei	David	0722123456	02-JUL-20	3250	601 PAZNIC	8
17	20320	Popescu	Stefan	0721145456	02-JUL-20	3400	700 PAZNIC	9
18	20340	Ion	Catalin	0721765456	02-JUL-20	3320	601 PAZNIC	10

Script Output | Query Result | Query Result

SQL | All Rows Fetched: 10 in 0.002 seconds

ID_ECHIPA_PAZA	ID_FIRMA
1	601
2	601
3	800
4	700
5	900
6	1000
7	1000
8	601
9	700
10	601

The screenshot shows the Oracle SQL Developer interface with a PL/SQL script editor and a results tab.

PL/SQL Script Editor:

```
1  declare
2    begin
3      if :v1 < :v2 then
4          dbms_output.put_line('Data de plecare este mai mare decat data de sosire');
5      else
6          dbms_output.put_line('Data de plecare este mai mica decat data de sosire');
7      end if;
8  end;
```

Results Tab:

```
1 row inserted.

1 row inserted.
```

Bottom Status Bar:

Compiler Log: Click on an identifier with the Control key down to perform 'Go to Declaration'.

The screenshot shows the Oracle SQL Developer interface with a running PL/SQL script. The code is as follows:

```
1-- Oracle SQL Developer - project
2
3-- Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
4-- Version 19.1.0.0.0
5
6-- Copyright (c) 1982, 2019, Oracle. All rights reserved.
7-- Licensed Material - Property of Oracle
8
9-- 100% Complete
10-- 1 row updated.
11
12-- 100% Complete
13-- 1 row updated.
14
15-- 100% Complete
16-- 1 row updated.
17
18-- 100% Complete
19-- 1 row updated.
20
21-- 100% Complete
22-- 1 row updated.
23
24-- 100% Complete
25-- 1 row updated.
26
27-- 100% Complete
28-- 1 row updated.
29
30-- 100% Complete
31-- 1 row updated.
32
33-- 100% Complete
34-- 1 row updated.
35
36-- 100% Complete
37-- 1 row updated.
38
39-- 100% Complete
40-- 1 row updated.
41
42-- 100% Complete
43-- 1 row updated.
44
45-- 100% Complete
46-- 1 row updated.
47
48-- 100% Complete
49-- 1 row updated.
50
51-- 100% Complete
52-- 1 row updated.
53
54-- 100% Complete
55-- 1 row updated.
56
57-- 100% Complete
58-- 1 row updated.
59
60-- 100% Complete
61-- 1 row updated.
62
63-- 100% Complete
64-- 1 row updated.
65
66-- 100% Complete
67-- 1 row updated.
68
69-- 100% Complete
70-- 1 row updated.
71
72-- 100% Complete
73-- 1 row updated.
74
75-- 100% Complete
76-- 1 row updated.
77
78-- 100% Complete
79-- 1 row updated.
80
81-- 100% Complete
82-- 1 row updated.
83
84-- 100% Complete
85-- 1 row updated.
86
87-- 100% Complete
88-- 1 row updated.
89
90-- 100% Complete
91-- 1 row updated.
92
93-- 100% Complete
94-- 1 row updated.
95
96-- 100% Complete
97-- 1 row updated.
98
99-- 100% Complete
100-- 1 row updated.
```

The script consists of several UPDATE statements for tables like 'istoric_popasuri' and 'istoric_sosire'. It includes WHERE clauses to filter data based on dates and triggers. The execution progress bar at the bottom indicates 100% completion for each row update. The status bar at the bottom left shows '100% Complete' and '1 row updated.'

The screenshot shows the Oracle SQL Developer interface with a script editor window open. The code in the editor is as follows:

```
1--> 1 --> 2--> 3--> 4--> 5--> 6--> 7--> 8--> 9--> 10--> 11--> 12--> 13--> 14--> 15--> 16--> 17--> 18--> 19--> 20--> 21--> 22--> 23--> 24--> 25--> 26--> 27--> 28--> 29--> 30--> 31--> 32--> 33-->
```

Execution results are displayed below the code:

```
Table AUDIT_USER created.

Trigger AUDIT_SCHEMA compiled

Table TEST created.

Table TEST altered.

Name      Null? Type
-----  -----
ID          VARCHAR2(25)
COLONA_NOGA  VARCHAR2(25)

Index INDEX_TEST created.

Index INDEX_TEST dropped.

Table TEST dropped.

Type O compiled

Type O dropped.
```

```

11  --> AFTER CREATE OR DROP OR ALTER ON SCHEMA
12  BEGIN
13    INSERT INTO audit_user
14      VALUES (SYS_DATABASE_NAME, SYS.LOGIN_USER,
15              SYS.SYSEVENT, SYS.DICTIONARY_OBJ_TYPE,
16              SYS.DICTIONARY_OB_NAME, SYSTIMESTAMP());
17  END;
18  /
19
20  CREATE TABLE test (id varchar2(25));
21  ALTER TABLE test ADD colcana_noua varchar2(25);
22
23  desc test;
24
25  CREATE INDEX index_test ON test(id);
26  DROP INDEX index_test;
27
28  DROP Table test;
29
30  CREATE OR REPLACE TYPE o IS OBJECT (t varchar(2));
31  DROP TYPE o;
32
33  select * from audit_user;

```

View Object > Query Result

NAME_ID	USER_ID	EVENT_ID	TP_OBJECT_ID	NAME_OBJECT	DATA
1XXE	PROJECT	CREATE TABLE		TEST	30-DEC-21 01.08.39.266000000 AM
1XXE	PROJECT	DROP TABLE		TEST	30-DEC-21 01.08.46.111000000 AM
1XXE	PROJECT	CREATE INDEX		INDEX_TEST	30-DEC-21 01.08.52.062000000 AM
1XXE	PROJECT	DROP INDEX		INDEX_TEST	30-DEC-21 01.08.53.787000000 AM
1XXE	PROJECT	CREATE TABLE		TEST	30-DEC-21 01.08.55.000000000 AM
1XXE	PROJECT	CREATE TYPE		o	30-DEC-21 01.08.57.699000000 AM
1XXE	PROJECT	DROP TYPE		o	30-DEC-21 01.08.59.140000000 AM

```

424
425  EXCEPTION
426    WHEN NO_DATA_FOUND THEN -- pentru cazul in care nu am gasit transportul
427      DBMS_OUTPUT.PUT_LINE('Camionul ('||p_nr_immatriculare||'
428                            ||' nu are nicio marfa in data de'||p_data_piese);
429
430    WHEN TOO_MANY_ROWS THEN -- pentru cazul cand se adauga mai multe transporturi intr-o zi
431      DBMS_OUTPUT.PUT_LINE('Camionul ('||p_nr_immatriculare||'
432                            ||' a realizat mai multe transporturi in data de'||p_data_piese);
433
434    WHEN transport_fara_marfa THEN -- daca am gasit transportul dar nu si marfa
435      DBMS_OUTPUT.PUT_LINE('Camionul ('||p_nr_immatriculare||'
436                            ||' a realizat un transport in data de'||p_data_piese);
437
438    WHEN OTHERS THEN -- altceva este nevoie sa returnam marfa transportata. Transport invalid.')
439      DBMS_OUTPUT.PUT_LINE(SQLEERR);
440  END;
441  END pack_ex13;
442  /
443
444  -- testam o procedura din pacchet
445  EXECUTE pack_ex13.tracking_proc;

```

Procedure pack_ex13 completed in 0.12 seconds

Package PACK_EX13 compiled

Package Body PACK_EX13 compiled

Camion cu marca VOLVO avand nr. de immatriculare CT12FSD si este detinut de compania Road Logistics.
Status transporturi:
Transport in data de 06-05-2021 realizat de Ion Vasile care este platit lunar cu 5000lei.
Transport in data de 11-05-2021 realizat de Ion Vasile care este platit lunar cu 5000lei.

Camionul cu marca VOLVO avand nr. de immatriculare CT98000M si este detinut de compania Road Logistics.
Status transporturi:
Transport in data de 10-05-2021 realizat de Ovidiu Ion care este platit lunar cu 4000lei.
Transport in data de 10-05-2021 realizat de Ovidiu Ion care este platit lunar cu 4000lei.

Camionul cu marca SCANIA avand nr. de immatriculare PHI38CN si este detinut de compania Transibio.
Status transporturi:
Transport in data de 06-05-2021 realizat de Marian Gheorghe care este platit lunar cu 3000lei.

Camionul cu marca DAF avand nr. de immatriculare B55ASD si este detinut de compania EasyCargo.
Status transporturi:
Transport in data de 06-05-2021 realizat de Manea George care este platit lunar cu 4000lei.

Camionul cu marca MAN avand nr. de immatriculare TM200WE si este detinut de compania Road Logistics.
Status transporturi:
Nu au fost realizate transporturi cu acest camion.

