

Modelul de date va gestiona activitatea unei companii de transport de mărfuri din România. În cadrul firmei există mai multe puncte de lucru denumite în continuare “Departamente”, în diferite orașe din România. Activitatea firmei o reprezintă transportul de marfă, realizat prin cursele șoferilor angajați.

Fiecare departament este condus de către un manager care încheie contracte cu diverși beneficiari. Pe baza acestor contracte se realizează cursele.

Un autovehicul poate fi condus de mai mulți șoferi. Acest autovehicul este implicat în curse și trebuie să meargă la revizii.

Reguli:

- Pentru ca un contract să poată fi încheiat, trebuie să fie disponibil cel puțin un autovehicul capabil să transporte cantitatea și tipul mărfii din contract.
- Un autovehicul nu este disponibil pentru cursă într-o zi, dacă acesta iese de la revizie în acea zi.
- Când este cumpărat un autovehicul, acesta este supus unei revizii generale obligatorii.
- Un autovehicul poate fi condus de maxim 2 șoferi în aceeași cursă (unul dintre aceștia conduce ziua, iar celălalt noaptea sau formează o echipă pentru a nu sta pe loc în timpul de odihnă).
- Dacă se încheie un contract pentru marfă peste 22 de tone, se vor grupa mai multe camioane, acestea plecând într-o singură cursă împreună (cursa va avea același ID pentru toate camioanele).
- Cheltuielile din curse se calculează în felul următor: $\text{km} * (\text{consum}/100) * 4$. Consumul poate varia în funcție de cât de plin este autovehiculul, iar 4 semnifică prețul (special) pe litru al motorinei.

Constrângeri:

Entități

Pentru modelul de date referitor la transportul de marfă, structurile DEPARTAMENT, LOCATIE, ANGAJATI, MANAGER, SOFER, AUTOVEHICUL, CURSA, REVIZII, CONTRACT, ACCIDENTE, BENEFICIAR, reprezintă entități.

Toate entitățile prezentate sunt independente, cu excepția subentităților SOFER și MANAGER.

DEPARTAMENT = punct de lucru în care lucrează angajații. Cheia primară a acestei entități este *dept_id*.

LOCATIE = entitate care identifică locația unui departament. Cheia primară a acestei entități este *locatie_id*.

ANGAJATI = persoană fizică, angajată în cadrul companiei, fie pe poziție de conducere și pentru a încheia contracte (manager), fie pentru a transporta marfă cu un autovehicul (șofer). Cheia primară a acestei entități este *angajat_id*.

MANAGER = subentitate a entității ANGAJATI, ce conține informații specifice postului ocupat. Cheia primară a acestei entități este *angajat_id*.

SOFER = subentitate a entității ANGAJATI, ce conține informații specifice conducătorilor auto. Cheia primară a acestei entități este *angajat_id*.

AUTOVEHICUL = autovehiculul condus de către șofer pentru a transporta marfă. Cheia primară a acestei entități este *nr_inmatriculare*.

CURSA = activitatea principală a companiei. Modelul de date consideră toate cursele efectuate de către șoferi cu autovehiculele lor, la nivelul întregii companii (național). Cheia primară a acestei entități este *cursa_id*.

REVIZII = mentenanța autovehiculelor, efectuată periodic sau la nevoie. Cheia primară a acestei entități este *revizie_id*.

ACCIDENTE = accidente în care sunt implicate autovehiculele companiei. Cheia primară a acestei entități este *accident_id*.

CONTRACT = documentul care stabilește detaliile unei curse. Cheia primară a acestei entități este *contract_id*.

BENEFICIAR = persoana juridică ce solicită serviciile companiei. Cheia primară a acestei entități este *beneficiar_id*.

Relații

DEPARTAMENT *_se_afla_* LOCATIE = relația dintre DEPARTAMENT și LOCATIE, reflectă legătura dintre acestea (ce departament se află într-o locație). Deoarece într-o locație se poate afla un singur departament, cardinalitatea minimă este 1:1, și cardinalitatea maximă este 1:1.

ANGAJATI *_lucreaza_in_* DEPARTAMENT = relație ce leagă entitățile ANGAJATI și DEPARTAMENT (apartenența salariaților la un departament). Relația are cardinalitatea minimă 1:1 și cardinalitatea maximă 1:n.

MANAGER *_incheie_* CONTRACT = relația leagă entitățile MANAGER și CONTRACT (ce manager încheie un anumit contract). Relația are cardinalitatea minimă 1:0 și cardinalitatea maximă 1:n.

BENEFICIAR *_incheie_* CONTRACT = relația leagă entitățile BENEFICIAR și CONTRACT (cu ce beneficiar este încheiat un anumit contract). Relația are cardinalitatea minimă 1:1 și cardinalitatea maximă 1:n.

SOFER *_pleaca_in_* CURSA *_cu_* AUTOVEHICUL = relație de tip 3 ce elagă entitățile SOFER, CURSA și AUTOVEHICUL, reflectând cine a plecat în ce cursă și cu ce autovehicule. Denumirea acestei relații va fi *plecare*.

CURSA *_corespunde_* CONTRACT = relația leagă entitățile CURSA și CONTRACT (ce cursa corespunde unui contract). Deoarece o cursă se realizează în baza unui unic contract, relația are cardinalitatea minimă 1:1 și cardinalitatea maximă 1:1.

AUTOVEHICUL *_merge_la_* REVIZII = relația leagă entitățile AUTOVEHICUL și REVIZII (ce autovehicule merg la revizie). Dat fiind faptul că **orice** autovehicul trebuie supus unei verificări (revizii) după ce intră în companie, relația are cardinalitatea minimă 1:1 și cardinalitatea maximă 1:n.

AUTOVEHICUL *_implicat_in_* ACCIDENTE = relația de tip *many_to_many* dintre AUTOVEHICUL și ACCIDENTE (ce autovehicule sunt implicate în accidente). Relația are cardinalitatea minimă 1:0 și cardinalitatea maximă *m:n*.

Atribute

Entitatea independentă LOCATIE are ca atribute:

locatie_id = variabilă de tip întreg, de lungime maximă 4, care reprezintă codul locației.

judet = variabilă de tip caracter, de lungime maximă 20, care reprezintă numele județului.

oras = variabilă de tip caracter, de lungime maximă 30, care reprezintă numele orașului.

strada = variabilă de tip caracter, de lungime maximă 50, care conține numele străzii.

Entitatea independentă ANGAJATI are ca atribute:

angajat_id = variabilă de tip întreg de lungime maximă 4, care reprezintă codul unui angajat.

nume = variabilă de tip caracter, de lungime maximă 25, care reprezintă numele unui angajat

prenume = variabilă de tip caracter, de lungime maximă 25, care reprezintă prenumele unui angajat.

salariu = variabilă de tip întreg, de lungime maximă 5, care reprezintă salariul în lei al unui angajat.

dept_id = variabilă de tip întreg, de lungime maximă 4, care reprezintă codul departamentului în care lucrează angajatul.

Subentitatea MANAGER are ca atribute:

angajat_id = variabilă de tip întreg de lungime maximă 4, care reprezintă codul unui manager.

comision = variabilă de tip numeric (real), care reprezintă comisionul câștigat de manager sub formă zecimală.

Entitatea CONTRACT are ca attribute:

contract_id = variabilă de tip întreg, de lungime maximă 20, care reprezintă codul unui contract

manager_id = variabilă de tip întreg, de lungime maximă 20, care reprezintă codul unui manager.

beneficiar_id = variabilă de tip întreg, de lungime maximă 3, care reprezintă codul unui beneficiar.

suma = variabilă de tip numeric(real), care reprezintă suma platită de către beneficiar.

cantitate = variabilă de tip numeric(real), care reprezintă cantitatea de marfă transportată în tone.

tip_marfa = variabilă de tip caracter, de lungime maximă 20, care reprezintă categoria de marfă transportată.(generală, temperatura controlată, alimentară, ADR)

oras_plecure = variabilă de tip caracter, de lungime maximă 30, care conține numele orașului din care începe cursa.

oras_sosire = variabilă de tip caracter, de lungime maximă 30, care conține numele orașului în care se termină cursa.

zi_plecure = variabilă de tip dată calendaristică, reprezentând data la care începe cursa.

zi_sosire = variabilă de tip dată calendaristică, reprezentând data la care se termină cursa.

Entitatea BENEFICIAR are ca attribute:

beneficiar_id = variabilă de tip întreg, de lungime maximă 3, care reprezintă codul unui beneficiar.

denumire = variabilă de tip caracter de lungime maximă 30, care reprezintă numele oficial al beneficiarului.

persoana_contact = variabilă de tip caracter, de lungime maximă 50, care reprezintă numele persoanei de contact a beneficiarului.

telefon = variabilă de tip întreg, de lungime maximă 10, care reprezintă numărul de telefon al persoanei de contact a beneficiarului.

Entitatea DEPARTAMENT are ca attribute:

dept_id = variabilă de tip întreg, de lungime maximă 4, care reprezintă codul departamentului.

locatie_id = variabilă de tip întreg, de lungime maximă 4, care reprezintă codul locației.

manager_id = variabilă de tip întreg, de lungime maximă 20, care reprezintă codul unui manager.

Entitatea SOFER are ca attribute:

angajat_id = variabilă de tip întreg, de lungime maximă 4, care reprezintă codul șoferului.

manager_id = variabilă de tip întreg, de lungime maximă 4, care reprezintă codul managerului unui șofer (șeful său).

experienta = variabilă de tip întreg, de lungime maximă 2, care reprezintă experiența unui șofer în ani.

Entitatea AUTOVEHICUL are ca attribute:

nr_inmatriculare = variabilă de tip întreg, de lungime maximă 11, care reprezintă numărul de înmatriculare al autovehiculului condus de șofer. Acest cod este format din prescurtarea județului, 2 sau 3 cifre (în cazul în care prima literă este 'B') și 3 litere (ex.: IF-26-FLV, B-263-FLV).

tip = variabilă de tip întreg, de lungime maximă 2, care reprezintă codul tipului de marfă transportat de autovehicul.

capacitate = variabilă de tip numeric(real), care reprezintă capacitatea autovehiculului.

utilitate = variabilă de tip caracter, care reprezintă numele unei utilități. (generală, temperatură controlată, alimentară, ADR).

consum_gol = variabilă de tip numeric(real), care reprezintă consumul la suta de kilometri, când autovehiculul nu transportă marfă.

consum_plin = variabilă de tip numeric(real), care reprezintă consumul la suta de kilometri, când autovehiculul transportă marfă la capacitatea maximă.

Entitatea CURSA are ca attribute:

cursa_id = variabilă de tip întreg, de lungime maximă 7, care reprezintă codul unei curse.

km = variabilă de tip numeric(real), care reprezintă numărul de kilometri parcuși în cursă.

contract_id = variabilă de tip întreg, de lungime maximă 20, care reprezintă codul contractului corespunzător cursei.

Entitatea REVIZII are ca attribute:

revizii_id = variabilă de tip întreg, de lungime maximă 4 care reprezintă codul unei revizii.

cost = variabilă de tip numeric(real), care reprezintă costul efectuării unei revizii.

data_in = variabilă de tip dată calendaristică, reprezentând data la care începe efectuarea reviziei.

data_out = variabilă de tip dată calendaristică, reprezentând data la care ia sfârșit revizia.

nr_inmatriculare = variabilă de tip întreg, de lungime maximă 11, care reprezintă numărul de înmatriculare al unui autovehicul.

Relația *AUTOVEHICUL _implicat_in_ ACCIDENTE* are ca atribute:

nr_inmatriculare = variabilă de tip întreg, de lungime maximă 11, care reprezintă numărul de înmatriculare al unui autovehicul. Atributul trebuie să corespundă la o valoare a cheii primare din tabelul AUTOVEHICULE.

accident_id = variabilă de tip întreg, de lungime maximă 4, care reprezintă codul unui accident. Atributul trebuie să corespundă la o valoare a cheii primare din tabelul ACCIDENTE.

Relația *SOFER _pleaca_in CURSA _cu_ AUTOVEHICUL* are ca atribute *angajat_id*, *cursa_id*, *nr_inmatriculare*, *cheltuieli*.

Diagrama entitate/relație

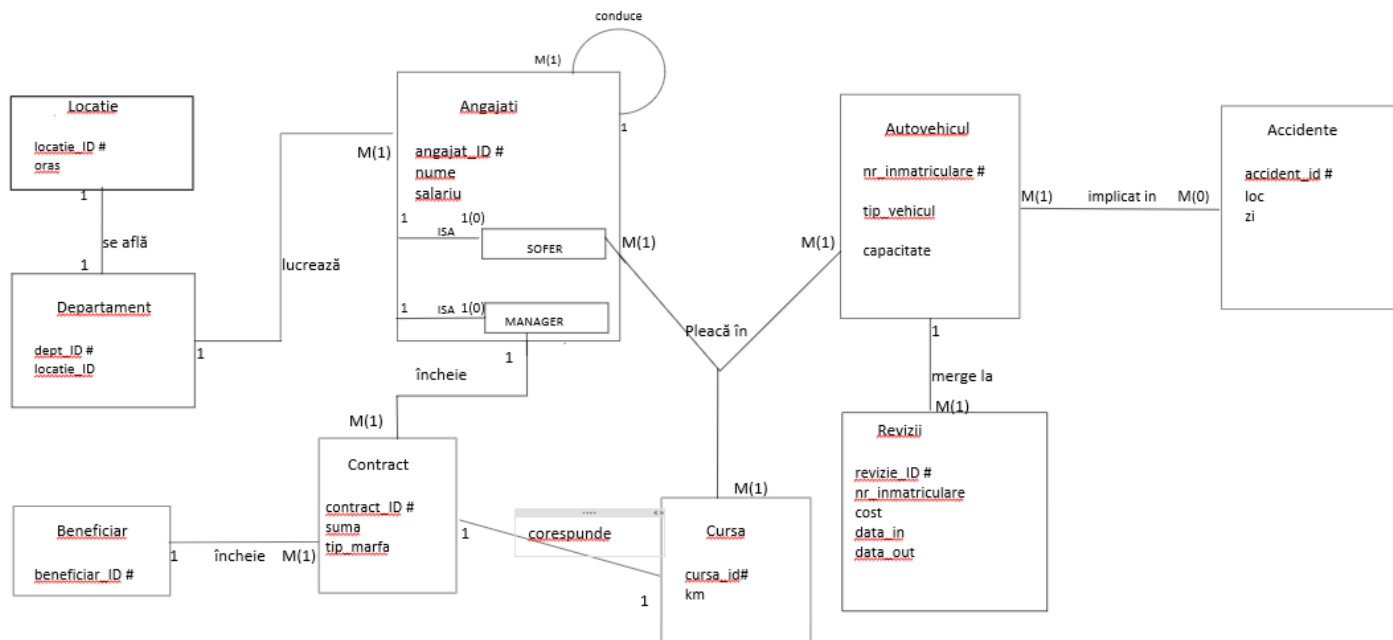
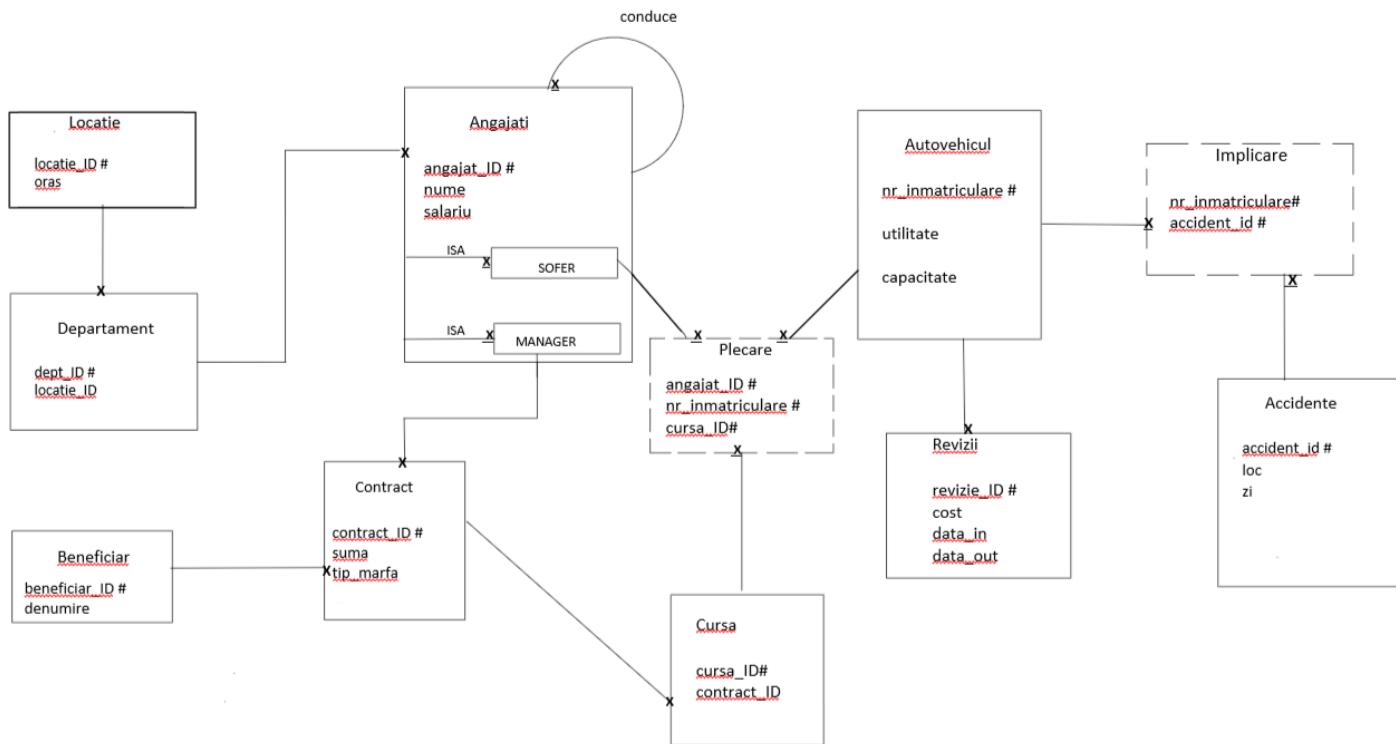


Diagrama conceptuală



Schema relațională

DEPARTAMENT(dept_ID#, locatie_ID, manager_ID);

LOCATIE(locatie_ID#, judet, oras, strada);

ANGAJATI(angajat_ID#, dept_ID, nume, prenume, salariu);

MANAGER(angajat_ID#, comision);

SOFER(angajat_ID#, manager_ID, experienta);

AUTOVEHICUL(nr_inmatriculare#, tip_vehicul, capacitate, utilitate, consum_gol, consum_plin);

CURSA(cursa_ID#, contract_ID#, km);

PLECARE(cursa_ID#, angajat_ID#, nr_inmatriculare#, cheltuieli);

REVIZII(revizie_ID#, cost, data_in, data_out);

ACCIDENTE(accident_ID#, loc, zi);

IMPLICARE(accident_ID#, nr_inmatriculare#);

CONTRACT(contract_ID#, manager_ID, suma, cantitate, tip_marfa, oras_plecure, oras_sosire, zi_plecure, zi_sosire);

BENEFICIAR(beneficiar_ID#, denumire, persoana_contact, telefon);

Normalizarea

-exemple -

Non FN1:

accident_ID#	nr_inmatriculare
100	'B-06-CYN', 'CT-34-JOP'
101	'B-602-MIS'
102	'BV-47-OLF', 'B-134-CVP'

Valorile din coloana 'nr_inmatriculare' nu sunt indivizabile.

FN1:

accident_ID	nr_inmatriculare#
100	'B-06-CYN'
100	'CT-34-JOP'
101	'B-602-MIS'
102	'BV-47-OLF'
102	'B-134-CVP'

Non FN2:

accident_ID#	nr_inmatriculare#	loc
100	'B-06-CYN'	bucuresti
100	'CT-34-JOP'	bucuresti
101	'B-602-MIS'	pitesti
102	'BV-47-OLF'	brasov
102	'B-134-CVP'	brasov

Locul accidentului nu depinde de masina, ci numai de codul accidentului.

FN2:

accident_ID#	nr_inmatriculare#
100	'B-06-CYN'
100	'CT-34-JOP'
101	'B-602-MIS'
102	'BV-47-OLF'
102	'B-134-CVP'

accident_ID#	loc
100	bucuresti
100	bucuresti
101	pitesti
102	brasov
102	brasov

Non FN3:

nr_inmatriculare	utilitate	capacitate	revizie_ID	cost
'BV-47-OLF'	ADR	7.5	502	50

Acest exemplu este non FN3 pentru că există o dependență tranzitivă:

cost depinde de *revizie_ID*, iar *revizie_ID* depinde de *nr_imatricualre*.

FN3:

nr_inmatriculare	utilitate	capacitate
'BV-47-OLF'	ADR	7.5
revizie_ID	nr_inmatriculare	cost
502	'BV-47-OLF'	50

Printscreens

11-13.sqlWelcome PageProiect BDANGAJATI

ColumnsDataModelConstraintsGrantsStatisticsTriggersFlashbackDependence

Sort..Filter:

	ANGAJAT_ID	NUME	PRENUME	SALARIU	DEPT_ID
1	100	Dumitru	Marian	2500	200
2	101	Popescu	Ion	3600	202
3	102	Chirita	Simona	12000	201
4	103	Podariu	Flavia	12500	200
5	104	Florea	Marius	5000	201
6	105	Barbu	Tudor	4780	203
7	106	Stan	Alexandru	3730	204
8	107	Craciun	Andrei	6100	202
9	108	Tabacu	Andreea	9720	203
10	109	Iancu	Cristian	8700	204
11	110	Farfurie	Gabriela	10000	202
12	111	Muntean	Sergiu	3000	201
13	112	Pop	George	3400	203

11-13.sqlWelcome PageProiect BD AUTOVEHICUL

ColumnsDataModelConstraintsGrantsStatisticsTriggersFlashbackDependenciesDetailsPartitionsIndexesSQL

Sort..Filter:

	NR_INMATRICULARE	UTILITATE	CAPACITATE	CONSUM_GOL	CONSUM_PLIN
1	B-602-MIS	generală	22	25	40
2	B-06-CYN	temperatura controlată	3.5	9.4	14.7
3	BV-47-OLF	ADR	7.5	16.2	24.9
4	CJ-76-FLM	generală	22	23.1	38.5
5	CT-34-JOP	temperatura controlată	22	26	39.9
6	B-134-CVP	ADR	15	15.6	26.3
7	CJ-88-SSD	generală	3.5	8.9	15.4

11-13.sql Welcome Page Proiect BD BENEFICIAR

Columns Data Model Constraints Grants Statistics Triggers Flashback Dependencies Details

Sort.. Filter:

	BENEFICIAR_ID	DENUMIRE	PERSOANA_CONTACT	TELEFON
1	400	Metro	Marius Alexandrescu	723123123
2	401	Good Catering	Teodora Pop	725678987
3	402	IKEA	David Panait	765445932
4	403	Linde	Radu Marin	743778443
5	404	Betty Ice	Felicia Dinescu	754123432
6	405	Carturesti	Vlad Nistor	745678123

11-13.sqlWelcome PageProiect BDCONTRACT

ColumnsDataModelConstraintsGrantsStatisticsTriggersFlashbackDependenciesDetailsPartitionsIndexesSQL

11-13.sql	Welcome Page	Proiect BD	CURSA					
Columns	Data	Model	Constraints	Grants	Statistics	Triggers	Flashback	Depend
						Sort..	Filter:	
	↕ CURSA_ID	↕ CONTRAC...	↕ KM					
1	1000	300	224					
2	1001	301	224					
3	1002	302	328					
4	1003	303	257					
5	1004	304	380					
6	1005	305	184					
7	1006	306	440					
8	1007	307	271					
9	1008	308	224					
10	1009	309	79					
11	1010	310	444					

	NR_IN...	SOFER_ID	CURSA_ID	CHELTUIELI
1	B-06-CYN	100	1009	46
2	B-06-CYN	104	1005	108
3	B-134-CVP	111	1001	197
4	B-602-MIS	100	1006	704
5	B-602-MIS	104	1010	710
6	B-602-MIS	100	1010	710
7	B-602-MIS	104	1008	305
8	B-602-MIS	111	1006	704
9	BV-47-OLF	107	1003	256
10	CJ-76-FLM	105	1004	585
11	CJ-76-FLM	112	1004	585
12	CJ-76-FLM	112	1010	684
13	CJ-76-FLM	105	1010	684
14	CJ-88-SSD	112	1002	131
15	CJ-88-SSD	105	1007	167
16	CJ-88-SSD	105	1002	131
17	CT-34-JOP	106	1000	305

	REVIZII_ID	COST	DATA_IN	DATA_OUT	NR_INMATRICUL
1	500	120	10-FEB-21	10-FEB-21	B-602-MIS
2	501	40	10-FEB-21	10-FEB-21	B-06-CYN
3	502	50	10-FEB-21	10-FEB-21	BV-47-OLF
4	503	120	10-FEB-21	10-FEB-21	CJ-76-FLM
5	504	120	10-FEB-21	10-FEB-21	CT-34-JOP
6	505	120	10-FEB-21	10-FEB-21	B-134-CVP
7	506	35	10-FEB-21	10-FEB-21	CJ-88-SSD

	ANGAJAT_ID	MANAGER_ID	EXPERIENTA
1	100	103	3
2	101	110	3
3	104	102	4
4	105	108	4
5	106	109	3
6	107	110	6
7	111	102	2
8	112	108	3

Printscreenuri ex 11

```
--1) Afisati numele, orasul in care s-au angajat toti soferii companiei, precum si numele managerului lor.
select initcap(a.num) as "Nume sofer",a.angajat_id as "ID sofer", initcap(l.oras) as "Oras", initcap(b.num) as "Nume manager",
       b.angajat_id as "ID Manager"
from angajati a, locatie l,sofer s, departament d, angajati b
where a.angajat_id = s.angajat_id -- angajatul sa fie sofer
and a.dept_id = d.dept_id -- join cu tabela de departament
and l.locatie_id = d.locatie_id -- datorita joinului de sus acum am gasit locatia angajatului
and b.angajat_id = s.manager_id -- b.angajat_id este managerul si trebuie sa fie acelasi cu managerul soferului
order by 1;
```

Nume sofer	ID sofer	Oras	Nume manager	ID Manager
1 Barbu	105	Cluj-Napoca	Tabacu	108
2 Craciun	107	Brasov	Farfurie	110
3 Dumitru	100	Bucuresti	Podariu	103
4 Florea	104	Bucuresti	Chirita	102
5 Muntean	111	Bucuresti	Chirita	102
6 Pop	112	Cluj-Napoca	Tabacu	108
7 Popescu	101	Brasov	Farfurie	110
8 Stan	106	Constanta	Iancu	109

```
--2) Afisati numele managerilor care au incheiat contracte asociate unor curse de cel putin 250 km in ultima luna.

select nume || ' ' || prenume as "Manager"
from angajati
where angajat_id in (select m.angajat_id
                     from manager m, contract c, cursa cr
                     where m.angajat_id = c.manager_id -- angajatul trebuie sa fie cel care incheie contractul
                     and c.contract_id = cr.contract_id -- join cu tabela de cursa ca sa pot gasi nr de km
                     and cr.km >= 250 -- conditia pt nr de km
                     and c.zi_plecure > add_months(sysdate, -1)); -- ziua de plecare sa fie mai mare decat data de acum o luna
```

Query Result x Query Result 1 x
 All Rows Fetched: 2 in 0.004 seconds

Manager
1 Podariu Flavia
2 Tabacu Andreea

```
--3) Afisati numele tuturor managerilor care castiga mai mult decat media salariilor cu bonus adaugat. Bonusul consta in
-- cresterea salariului vechi cu un procent echivalent cu comisionul castigat.
```

```
with templ(sal_mng) as (select avg(al.salariu + (al.salariu)* NVL(ml.comision, 0)) -- comisionul poate fi null
                        from angajati al, manager ml
                        where al.angajat_id = ml.angajat_id -- angajatul sa fie manager
                        )
select distinct initcap(a.num) || ' ' || initcap(a.prenume) as "Salariat", m.comision, a.salariu
from angajati a, manager m, sofer s, templ
where a.angajat_id = m.angajat_id --angajatul sa fie manager
and a.salariu > sal_mng -- se compara salariul cu cel calculat mai sus
order by
(
  case
    when m.comision is null then a.salariu -- ordonez rezultatele dupa comision daca acesta exista
    else m.comision -- altfel ordonez dupa salariul fara bonus
    end
);
```

Query Result x Query Result 1 x
 All Rows Fetched: 2 in 0.004 seconds

Salariat	COMISION	SALARIU
1 Podariu Flavia	0.1	12500
2 Chirita Simona	(null)	12000

```
--4) Afisati informatii despre autovehiculele de 22 de tone care au fost in curse mai lungi de 200km
-- si au consumul mediu mai mic decat media consumurilor medii ale tuturor autovehiculelor de 22 de tone.
```

```
with temp(cons_mediu) as (select avg((consum_gol + consum_plin)/2)
                           from autovehicul
                           where capacitate = 22
                           )
select distinct a.nr_inmatriculare, a.utilitate
from autovehicul a, temp
where a.capacitate = 22
and (a.consum_gol+a.consum_plin)/2 <= cons_mediu
and a.nr_inmatriculare in
(select p.nr_inmatriculare
 from plecarea p, cursa c
 where p.nr_inmatriculare = a.nr_inmatriculare -- masina selectata sa fie cea din plecarea curenta
 and p.cursa_id = c.cursa_id -- joinul cu tabela de curse
 and c.km > 200 -- conditia de numar de km
 );
```

Query Result x Query Result 1 x
 SQL | All Rows Fetched: 1 in 0.002 seconds

NR_INMATRICULARE	UTILITATE
1 CJ-76-FLM	generala

```
--5) Afisati utilitatea autovehiculelor si numarul de revizii efectuate la fiecare categorie de autovehicul, mai putin 'ADR'.
-- In plus, se va afisa si totalul in lei pentru fiecare tip de autovehicul.
```

```
select a.utilitate as "Tip auto", count(revizii_id) as "Numar revizii", decode(sum(cost), 0, 0, sum(cost)) as "Total - lei"
from revizii r, autovehicul a
where a.nr_inmatriculare = r.nr_inmatriculare
group by (a.utilitate)
having lower(a.utilitate) <> 'adr'
```

Query Result x Query Result 1 x
 SQL | All Rows Fetched: 2 in 0.003 seconds

Tip auto	Numar revizii	Total - lei
1 temperatura controlata	2	160
2 generala	3	275