

Proiect Baze de date

Informatica – Anul 1

Ilie Petre Cristian

Grupa 133

Descrierea modelului

Modelul de date realizeaza gestionarea informatiilor organizatorice si functionale ale unei scoli de soferi. Aceasta poate avea mai multe sedii in care lucreaza secretari si in care se tin lectiile teoretice. La aceste lectii participa atat un profesor de legislatie (angajat al scolii) cat si clienti. Acesti clienti pot participa la lectiile teoretice indiferent de sediul unde sunt sustinute, primind (in urma participarii) prezente. Clientul, in prezenta unui secretar, semneaza un contract cu scoala de soferi care poate fi platit in mai multe transe.

La aceasta firma mai sunt angajati si instructori auto ce realizeaza orele de pregatire practica. La aceste lectii participa un client si este utilizat un automobil detinut de firma. Dupa terminarea cursurilor de pregatire practica (precum si acumularea prezentelor anterior descrise) clientul poate sustine examenul teoretic iar ulterior, daca acesta promoveaza, poate participa si la examenul practic, scoala de soferi tinand cont de toate incercarile clientului in cauza.

Modelul de date respecta anumite reguli de functionare:

- O lectie teoretica predata implica un singur profesor de legislatie. La o lectie pot participa mai multi clienti.
- Orice ora de pregatire practica are asociat un singur client, un singur instructor auto si un singur automobil.
- Niciun angajat/client/sediu nu poate avea asociata mai mult de o adresa insa mai multi angajati/clienti/sedii pot impartasi aceeasi adresa.
- Un oras poate avea mai multe adrese asociate insa o adresa nu poate avea mai multe orase.
- Un secretar nu poate lucra la mai multe sedii insa un sediu poate avea mai multi secretari.
- Un contract nu poate fi realizat de mai multi secretari sau semnat de mai multi clienti. Un client poate semna mai multe contracte iar un secretar poate realiza mai multe contracte.
- Angajatii firmei pot fi instructori auto, profesori de legislatie sau secretari.
- Examenele sustinute de clienti pot fi teoretice sau practice.
- Pentru un contract se pot realiza mai multe plati.
- Instructorii auto si profesorii de legislatie apartin de firma, nu de un anumit sediu.
- Tipul de facultate absolvita de un Angajat de tipul „Profesor Legislatie” este obligatoriu sa fie Tehnic sau Juridic
- Categoria unui automobil introdus de firma este obligatoriu sa fie A1,A2,B1,BE sau B.

Descrierea entitatilor

Pentru modelul de date referitor la scoala de soferi, structurile *SEDIU*, *ANGAJAT*, *INSTRUCTOR*, *PROF_LEGIS*, *SECRETAR*, *LECTIE_PRACTICA*, *AUTOMOBIL*, *LECTIE_TEORETICA*, *CONTRACT*, *PLATA*, *ADRESA*, *CLIENT*, *EXAMEN*, *TEORETIC*, *PRACTIC*, *ORAS* reprezinta entitati.

SEDIU = una din locatiile unde se desfasoara activitatiile principale ale scolii de soferi cum ar fi inrolarea elevilor precum si instruirea teoretica a acestora. Cheia primara a entitatii este *id_sediu*.

ANGAJAT = persoana fizica, angajata permanent fie ca secretar, profesor de legislatie sau instructor auto. Cheia primara este *id_angajat*.

INSTRUCTOR = subentitate a entitatii *ANGAJAT*, ce contine in plus informatii asupra experientei in condus a angajatului. Cheia primara este *id_angajat*

PROF_LEGIS = subentitate a entitatii *ANGAJAT*, ce contine informatii aditionale asupra tipul facultatii absolvite, data promovarii examenului de atestare profesionala. Cheia primara este *id_angajat*.

SECRETAR = subentitate a entitatii *ANGAJAT*, ce contine informatii asupra sediului in care isi desfasoara activitatea profesionala. Cheia primara este *id_angajat*.

LECTIE_PRACTICA = ora de pregătire practică realizată de un instructor auto, la care este utilizat un automobil și la care participă un elev. Cheia primară este *id_lectiepr*.

AUTOMOBIL = vehicul deținut de firmă ce este utilizat pentru instruirea elevilor în cadrul orelor de pregătire practică. Se consideră că un vehicul nu este atașat unui singur instructor auto astfel încât aceștia să poată utiliza automobilul cel mai apropiat preferințelor clientului. Cheia primară este *id_automobil*.

LECTIE_TEORETICA = ora de pregătire teoretică ce este realizată de către un profesor de legislație rutieră în cadrul uneia dintre sediile firmei. Cheia primară este *id_lectietr*.

CONTRACT = act ce atestă înrolarea unei persoane fizice la școală de șoferi. Acesta este realizat de către un secretar al școlii și semnat de un client. Acest contract presupune plata unei sume de bani care poate fi realizată în mai multe tranșe. Cheia primară este *id_contract*.

PLATA = reprezintă dovada faptului că un client a achitat o sumă de bani pentru contractul semnat anterior de către acesta. Cheia primară este *id_plata*.

ADRESA = entitate ce reflectă fie domiciliul unei persoane (în cazul unui client/secretar) sau adresă la care se poate găsi unul din sediile firmei. Cheia primară este *id_adresa*.

CLIENT = persoana fizica ce semneaza un contract cu scoala de soferi pentru a fi inrolat si a realiza o pregatire atat teoretica cat si practica in detrimentul obtinerii carnetului de sofer. Cheia primara este id_client.

EXAMEN = entitatea face referire la un mijloc de verificare al cunostiintelor acaparate de-alungul scolii de soferi a unui client. Entitatea retine atat data la care a fost dat cat si daca clientul a fost Admis sau Respins. Cheia primara este id_examen.

TEORETIC = subentitate a entitatii EXAMEN, ce face referire la partea de teorie a examenului oficial in vederea obtinerii carnetului de sofer si ce contine in plus informatii asupra punctajului (intrebari la care a fost raspuns corect) obtinut de catre clienti. Cheia primara este id_examen.

PRACTIC = subentitate a entitatii EXAMEN, face referire la partea practica (traseul) a examenului. Aceasta contine in plus date asupra punctelor de penalizare acumulate de-alungul examenului practic. Cheia primara este id_examen.

ORAS = entitate ce reprezinta orasele in care se pot situa clientii/sediile/angajatii firmei. Cheia primara este id_oras.

Descrierea relatiilor

SEDIU_lucreaza_SECRETAR = relatie ce leaga entitatile SEDIU si SECRETAR, reflectand legatura dintre acestea (ce secretari din firma lucreaza la un anumit sediu) . Relatia de cardinalitate minima este 1:1 si cardinalitate maxima 1:n.

SEDIU_tine_LECTIE_TEORETICA = relatie ce leaga entitatile SEDIU si LECTIE_TEORETICA, reflectand legatura dintre acestea (ce lectii de teorie sunt tinute la un anumit sediu) . Ea are cardinalitatea minima 1:0 si cardinalitatea maxima 1:n.

ADRESA_se_afla_SEDIU = relatie ce leaga entitatile ADRESA si SEDIU , reflectand legatura dintre acestea (ce sedii se afla la o anumita adresa) . Relatia de cardinalitate minima este 1:0 si cardinalitate maxima 1:n.

ADRESA_locuieste_ANGAJAT = relatie ce leaga entitatile ADRESA si ANGAJAT, reflectand legatura dintre acestea (ce angajati au domiciliul la o anumita adresa) . Relatia de cardinalitate minima este 1:0 si cardinalitate maxima 1:n.

ADRESA_locuieste_CLIENT = relatie ce leaga entitatile ADRESA si CLIENT, reflectand legatura dintre acestea (ce clienti au domiciliul la o anumita adresa) . Relatia de cardinalitate minima este 1:0 si cardinalitate maxima 1:n.

ORAS_are_ADRESA = relatie ce leaga entitatile ORAS si ADRESA, reflectand legatura dintre acestea (ce adrese se afla intr-un oras) . Relatia de cardinalitate minima este 1:1 si cardinalitate maxima 1:n.

CLIENT_sustine_EXAMEN = relatie ce leaga entitatile CLIENT si EXAMEN, reflectand legatura dintre acestea (ce examene au fost sustinute de un client) . Cardinalitatea minima a acestei relatii este 1:0 iar maxima 1:n.

CLIENT_semneaza_CONTRACT = relatie ce leaga entitatile CLIENT si CONTRACT, reflectand legatura dintre acestea (ce contracte au fost semnate de un client). Relatia are o cardinalitate minima de 1:1 si o cardinalitate maxima 1:n.

CLIENT_participa_LECTIE_PRACTICA = relatie ce leaga entitatile CLIENT si LECTIE_PRACTICA, reflectand legatura dintre acestea (la ce lectii practice a participat clientul curent). Relatia are o cardinalitate minima de 1:0 si o cardinalitate maxima de 1:n.

CLIENT_prezenta_LECTIE_TEORETICA = relatie de tip Many-To-Many dintre entitatile CLIENT si LECTIE_TEORETICA, reflectand legatura dintre acestea (ce clienti sunt prezenti la lectiile de teorie). Relatia are cardinalitatea minima 1:0 si cardinalitatea maxima n:m.

AUTOMOBIL_este_folosit_LECTIE_PRACTICA = relatie ce leaga entitatile AUTOMOBIL si LECTIE_PRACTICA, reflectand legatura dintre acestea (la ce lectii practice a fost folosit un anumit autovehicul). Cardinalitatea minima a relatiei este 1:0 iar cardinalitatea maxima este 1:n.

INSTRUCTOR_tine_LECTIE_PRACTICA = relatie ce leaga entitatile INSTRUCTOR si LECTIE_PRACTICA, reflectand legatura dintre ele (ce lectii sunt predate de un instructor auto). Relatia de cardinalitate minima este 1:0 iar cea de cardinalitate maxima este 1:n.

PROF_LEGIS_preda_LECTIE_TEORETICA = relatie ce leaga entitatile PROF_LEGIS si LECTIE_TEORETICA, reflectand legatura acestora (ce cursuri de teorie sunt predate de un profesor de legislatie rutiera). Cardinalitatea minima a acestei relatii este 1:0 iar cardinalitatea maxima este 1:n.

CONTRACT_se_face_PLATA = relatie ce leaga entitatile CONTRACT si PLATA, reflectand legatura dintre acestea (ce plati au fost facute pentru un contract). Relatia de cardinalitate minima este 1:0 iar cea de cardinalitate maxima este 1:n.

SECRETAR_realizeaza_CONTRACT = relatie ce leaga entitatile SECRETAR si CONTRACT, reflectand legatura dintre acestea (ce contracte au fost realizate de secretar). Relatia de cardinalitate minima este 1:0 iar cea de cardinalitate maxima este 1:n.

Descrierea atributelor

Entitatea ANGAJAT are ca attribute:

- id_angajat = variabila de tip intreg, de lungime maxima 5, care reprezinta codul unui angajat
- nume = variabila de tip caracter, de lungime maxima 30, care reprezinta numele angajatului
- prenume = variabila de tip caracter, de lungime maxima 30, care reprezinta prenumele angajatului
- id_adresa = variabila de tip intreg, de lungime maxima 5, care reprezinta codul adresei unui angajat. Atributul trebuie sa corespunda la o valoare a cheii primare din tabelul ADRESA
- nr_telefon = variabila de tip caracter, de lungime maxima 15, care reprezinta numarul de telefon al unui angajat
- email = variabila de tip caracter, de lungime maxima 320, care reprezinta emailul unui angajat. Numarul maxim de caractere de 320 provine din faptul ca partea locala (stanga lui @) poate avea maxim 64 de caractere iar partea de domeniu (dreapta lui @) poate avea 255.
- cnp = variabila de tip caracter, de lungime maxima 13, care reprezinta codul numeric personal al unui angajat

Subentitatea INSTRUCTOR are ca attribute:

- id_angajat = variabila de tip intreg, de lungime maxima 5, care reprezinta codul instructorului
- data_carnet = variabila de tip data calendaristica, care reprezinta data in care instructorul a obtinut carnetul de sofer

Subentitatea PROF_LEGIS are ca attribute:

- id_angajat = variabila de tip intreg, de lungime maxima 5, care reprezinta codul profesorului de legislatie
- tip_fac = variabila de tip caracter, de lungime maxima 25, care reprezinta tipul de facultate pe care a absolvit-o profesorul. Atributul tip_fac poate lua valorile Tehnic sau Juridic.
- data_exam = variabila de tip data calendaristica, care reprezinta data la care profesorul a promovat examenul de atestare profesionala.

Subentitatea SECRETAR are ca attribute:

- id_angajat = variabila de tip intreg, de lungime maxima 5, care reprezinta codul secretarului
- id_sediu = variabila de tip intreg, de lungime maxima 5, care reprezinta codul sediului in care lucreaza secretarul. Atributul id_sediu trebuie sa corespunda la o valoare a cheii primare din tabelul SEDIU

Entitatea SEDIU are ca attribute:

- id_sediu = variabila de tip intreg, de lungime maxima 5, care reprezinta codul sediului
- id_adresa = variabila de tip intreg, de lungime maxima 5, care reprezinta codul adresei la care se afla sediul. Atributul id_adresa trebuie sa corespunda la o valoare a cheii primare din tabelul ADRESA

Entitatea ADRESA are ca attribute:

- id_adresa = variabila de tip intreg, de lungime maxima 5, care reprezinta codul adresei
- id_oras = variabila de tip intreg, de lungime maxima 5, care reprezinta codul orasului in care se afla adresa. Atributul id_oras trebuie sa corespunda la o valoare a cheii primare din tabelul ORAS
- cod_postal = variabila de tip intreg, de lungime maxima 6, care reprezinta codul postal
- info_adresa = variabila de tip caracter, de lungime maxima 60, care reprezinta informatiile aferente adresei (Numele strazii, Numarul strazii, Numarului blocului etc.)

Entitatea ORAS are ca attribute:

- id_oras = variabila de tip intreg, de lungime maxima 5, care reprezinta codul orasului
- denumire = variabila de tip caracter, de lungime maxima 50, care reprezinta denumirea orasului

Entitatea CLIENT are ca attribute:

- id_client = variabila de tip intreg, de lungime maxima 5, care reprezinta codul clientului
- nume = variabila de tip caracter, de lungime maxima 30, care reprezinta numele clientului
- prenume = variabila de tip caracter, de lungime maxima 30, care reprezinta prenumele clientului
- data_nasterii = variabila de tip data calendaristica, care reprezinta data de nastere a clientului
- id_adresa = variabila de tip intreg, de lungime maxima 5, care reprezinta codul adresei clientului. Atributul id_adresa trebuie sa corespunda la o valoare a cheii primare din tabelul ADRESA

- nr_telefon = variabila de tip caracter, de lungime maxima 15, care reprezinta numarul de telefon al unui client
- email = variabila de tip caracter, de lungime maxima 320, care reprezinta emailul unui client.
- cnp = variabila de tip caracter, de lungime maxima 13, care reprezinta codul numeric personal al unui client

Entitatea EXAMEN are ca attribute:

- id_examen = variabila de tip intreg, de lungime maxima 5, care reprezinta codul examenului
- id_elev = variabila de tip intreg, de lungime maxima 5, care reprezinta codul clientului ce sustine examenul. Atributul id_elev trebuie sa corespunda la o valoare a cheii primare din tabelul CLIENT
- data = variabila de tip data calendaristica, care reprezinta data la care s-a sustinut examenul
- calificativ = variabila de tip caracter, de lungime maxima 10, care reprezinta rezultatul examenului. Atributul calificativ poate lua valorile Admis sau Respins

Subentitatea TEORETIC are ca attribute:

- id_examen = variabila de tip intreg, de lungime maxima 5, care reprezinta codul examenului teoretic
- punctaj = variabila de tip intreg, de lungime maxima 2, care reprezinta numarul de intrebari la care a fost raspuns corect in cadrul examenului

Subentitatea PRACTIC are ca attribute:

- id_examen = variabila de tip intreg, de lungime maxima 5, care reprezinta codul examenului practic

- puncte_penalizare = variabila de tip intreg, de lungime maxima 3, care reprezinta codul examenului

Entitatea CONTRACT are ca attribute:

- id_contract = variabila de tip intreg, de lungime maxima 5, care reprezinta codul contractului
- id_secretar = variabila de tip intreg, de lungime maxima 5, care reprezinta codul secretarului care a realizat contractul. Atributul id_secretar trebuie sa corespunda la o valoare a cheii primare din tabelul SECRETAR
- id_client = variabila de tip intreg, de lungime maxima 5, care reprezinta codul clientului ce a semnat contractul. Atributul id_client trebuie sa corespunda la o valoare a cheii primare din tabelul CLIENT
- data_semnare = variabila de tip data calendaristica, reprezinta data la care a fost semnat contractul
- detalii_contract = variabila de tip caracter, de lungime maxima 255, care reprezinta detaliile contractului (Tip de permis, Automobil utilizat, Pret etc.)

Entitatea PLATA are ca attribute:

- id_plata = variabila de tip intreg, de lungime maxima 5, care reprezinta codul platii
- id_contract = variabila de tip intreg, de lungime maxima 5, care reprezinta codul contractului asupra caruia a fost realizata plata. Atributul id_contract trebuie sa corespunda la o valoare a cheii primare din tabelul CONTRACT
- data = variabila de tip data calendaristica, reprezinta data la care a fost facuta plata

- suma = variabila de tip intreg, de lungime maxima 10, care reprezinta suma de bani platita

Entitatea LECTIE_TEORETICA are ca attribute:

- id_lectietr = variabila de tip intreg, de lungime maxima 5, care reprezinta codul lectiei teoretice
- id_profesor = variabila de tip intreg, de lungime maxima 5, care reprezinta codul profesorului de legislatie care predă cursul. Atributul id_profesor trebuie sa corespunda la o valoare a cheii primare din tabelul PROF_LEGIS.
- id_sediu = variabila de tip intreg, de lungime maxima 5, care reprezinta codul sediului la care se desfasoara lectia. Atributul id_sediu trebuie sa corespunda la o valoare a cheii primare din tabelul SEDIU
- data = variabila de tip data calendaristica, care reprezinta data la care a fost tinuta lectia de pregatire teoretica
- detalii_suplimentare = variabila de tip caracter, de lungime maxima 150, care reprezinta o detaliere a subiectelor abordate in cadrul lectiei

Entitatea LECTIE_PRACTICA are ca attribute:

- id_lectiepr = variabila de tip intreg, de lungime maxima 5, care reprezinta codul lectiei practice
- id_instructor = variabila de tip intreg, de lungime maxima 5, care reprezinta codul instructorului care tine ora de instructaj. Atributul id_instructor trebuie sa corespunda la o valoare a cheii primare din tabelul INSTRUCTOR
- id_elev = variabila de tip intreg, de lungime maxima 5, care reprezinta codul clientului care participa la pregatirea practica.

Atributul id_elev trebuie sa corespunda la o valoare a cheii primare din tabelul CLIENT

- id_automobil = variabila de tip intreg, de lungime maxima 5, care reprezinta codul autovehiculului utilizat la ora de conducere. Atributul id_automobil trebuie sa corespunda la o valoare a cheii primare din tabelul AUTOMOBIL
- data = variabila de tip data calendaristica, care reprezinta data la care a fost realizata ora de instructaj practic
- km_parcursi = variabila de tip numeric (real), de lungime 5, care reprezinta numarul de kilometri parcursi in cadrul lectiei

Entitatea AUTOMOBIL are ca attribute:

- id_automobil = variabila de tip intreg, de lungime maxima 5, care reprezinta codul automobilului
- marca = variabila de tip caracter, de lungime maxima 30, care reprezinta marca autovehiculului
- model = variabila de tip caracter, de lungime maxima 30, care reprezinta modelul masinii
- categorie_automobil = variabila de tip caracter, de lungime maxima 2, care reprezinta categoria in care se clasifica autovehiculul. Aceasta atributa poate lua valorile A1,A2,B1,B,BE.

Relatia CLIENT_prezenta_LECTIE_TEORETICA are ca attribute:

- id_lectie = variabila de tip intreg, de lungime maxima 5, care reprezinta codul lectiei. Atributul id_lectie trebuie sa corespunda la o valoare a cheii primare din tabelul LECTIE_TEORETICA
- id_elev = variabila de tip intreg, de lungime maxima 5, care reprezinta codul clientului. Atributul id_elev trebuie sa corespunda la o valoare a cheii primare din tabelul CLIENT

Diagrama Entitate – Relatie

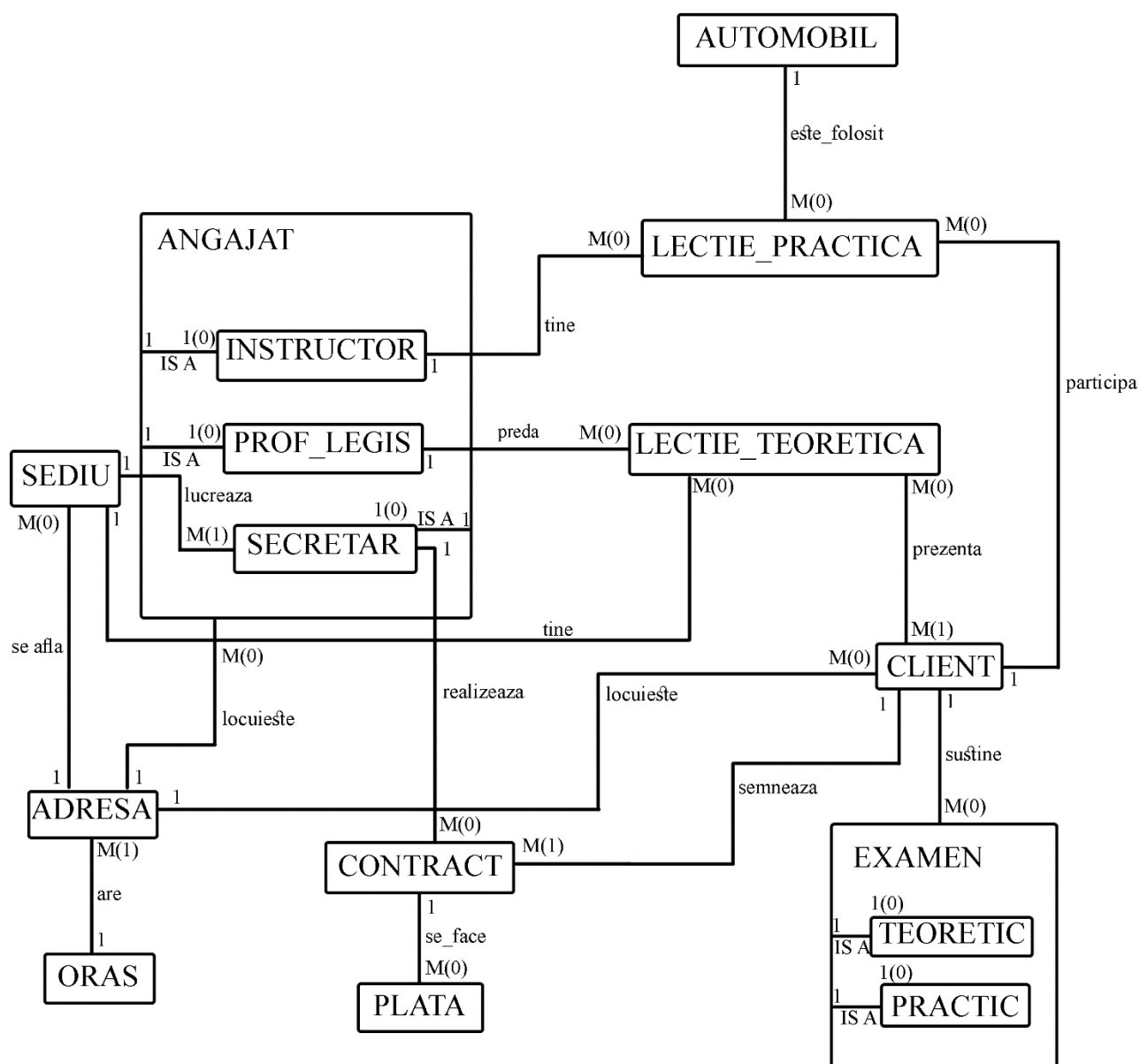
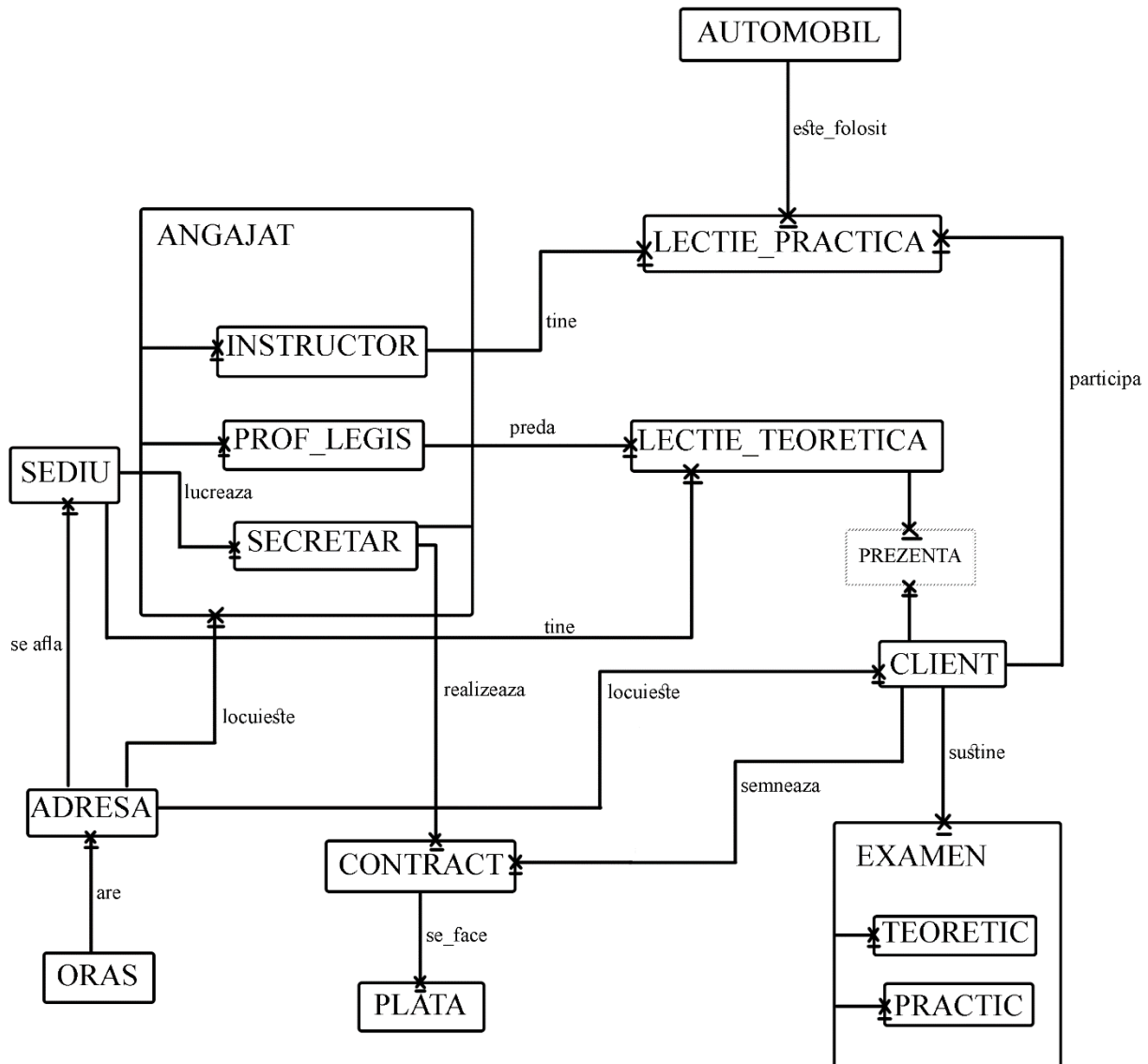


Diagrama Conceptuala



Scheme relationale

ANGAJAT(id_angajat#,nume,prenume,id_adresa,nr_telefon,email,cnp)

INSTRUCTOR(id_angajat#,data_carnet)

PROF_LEGIS(id_angajat#,tip_fac,data_exam)

SECRETAR(id_angajat#,id_sediu)

SEDIU(id_sediu#,id_adresa)

ADRESA(id_adresa#,id_oras,cod_postal,info_adresa)

ORAS(id_oras#,denumire)

CLIENT(id_client#,nume,prenume,data_nasterii,id_adresa,nr_telefon,email,cnp)

EXAMEN (id_examen#,id_elev,data,calificativ)

TEORETIC (id_examen#,punctaj)

PRACTIC (id_examen#,puncte_penalizare)

CONTRACT(id_contract#,id_secretar,id_client,data_semnare,detalii_contract)

PLATA (id_plata#,id_contract,data,suma)

LECTIE_TEORETICA

(id_lectietr#,id_profesor,id_sediu,data,detalii_suplimentare)

LECTIE_PRACTICA

(id_lectiepr#,id_instructor,id_elev,id_automobil,data,km_parcursi)

AUTOMOBIL (id_automobil#,marca,model,categorie_automobil)

PREZENTA(id_lectie#,id_elev#)

Normalizarea pana la FN3

Diagrama realizata anterior se afla deja in forma normala 3 si de aceea voi da cate un exemplu de transformare din Non-FN1 in FN1, din Non-FN2 in FN2 si din Non-FN3 in FN3.

Non-FN1

FN1

#id_oras	adresa
O1	A1,A3
O2	A2,A4



#id_oras	adresa
O1	A1
O1	A3
O2	A2
O2	A4

Non-FN2

FN2

#id_angajat	data_carnet	#id_elev
A1	25.02.2003	E2
A1	25.02.2003	E3
A2	13.06.2007	E1



#id_angajat	#id_elev
A1	E2
A1	E3
A2	E1

#id_angajat	Data_carnet
A1	25.02.2003
A2	13.06.2007

#id_angajat	Adresa_ang	Oras_ang
A1	Adr1	Oras1
A2	Adr2	Oras3
A3	Adr1	Oras1



#id_angajat	adresa
A1	Adr1
A2	Adr2
A3	Adr3





adresa	Oras
Adr1	O1
Adr2	O3
Adr3	O2

Crearea tabelelor in SQL

```
----- Creare tabele -----  
  
create table ORAS(  
  id_oras number(5) constraint id_oras_pk primary key,  
  denumire varchar2(50) constraint denumire_oras_nn not null  
);  
  
create sequence orase  
start with 1  
increment by 1;  
  
insert into ORAS  
values (orase.nextval, 'Bucuresti');  
insert into ORAS  
values (orase.nextval, 'Giurgiu');  
insert into ORAS  
values (orase.nextval, 'Popesti-Leordeni');  
insert into ORAS  
values (orase.nextval, 'Focsani');  
insert into ORAS  
values (orase.nextval, 'Buzau');  
  
select *  
from oras;
```

Script Output x

Query Result x

SQL | All Rows Fetched: 5 in 0.002 seconds

	ID_ORAS	DENUMIRE
1	1	Bucuresti
2	2	Giurgiu
3	3	Popesti-Leordeni
4	4	Focsani
5	5	Buzau

```

create table ADRESA(
id_adresa number(5) constraint id_adresa_pk primary key,
id_oras number(5) constraint adresa_oras_nn not null,
cod_postal number(6) constraint cod_postal_nn not null,
info_adresa varchar2(60),
foreign key (id_oras) references ORAS(id_oras)
);

create sequence adrese
start with 150
increment by 1;

insert into ADRESA (id_adresa,id_oras,cod_postal)
values (adrese.nextval,3,123456);
insert into ADRESA
values (adrese.nextval,4,512312,'Str. Stadionului, Nr. 45, Bl.2');
insert into ADRESA
values (adrese.nextval,5,278912,'Str. Gheorghe Doja Nr. 101');
insert into ADRESA
values (adrese.nextval,1,278912,'Str. Republicii Nr.1, Bl.18, Sc.2, Ap. 27');
insert into ADRESA
values (adrese.nextval,2,703503,'Str. Constantin Nottara, Bl. C/10, Ap. 3');
insert into ADRESA
values (adrese.nextval,2,408857,'Calea Eroilor nr. 70, bl. C, ap. 86');
insert into ADRESA
values (adrese.nextval,3,250633,'Calea Invatatorului 77, Mun. Cazanesti');

```

Script Output x Query Result x

SQL | All Rows Fetched: 15 in 0.004 seconds

	ID_ADRESA	ID_ORAS	COD_POSTAL	INFO_ADRESA
1	150	3	123456	(null)
2	151	4	512312	Str. Stadionului, Nr. 45, Bl.2
3	152	5	278912	Str. Gheorghe Doja Nr. 101
4	153	1	278912	Str. Constantin Nottara, Bl. C/10, Ap. 3
5	154	1	278912	Str. Republicii Nr.1, Bl.18, Sc.2, Ap. 27
6	155	2	408857	Calea Eroilor nr. 70, bl. C, ap. 86
7	156	3	250633	Calea Invatatorului 77, Mun. Cazanesti
8	157	1	777345	B-dul. Croitorilor 853, Mun. Piatra-Olt, Olt
9	158	4	429313	Calea J.J Rousseau nr. 75, bl. 8, ap. 02
10	159	5	732233	Calea Pădurii 9B, Sântana
11	160	5	236728	Splaiul Mircea cel Bătrân nr. 24, bl. B, ap. 96
12	161	4	205367	Str. Vlad Țepeș 3, Mun. Piatra-Olt, Alba
13	162	1	566457	Calea Someș 781
14	163	1	610487	Splaiul Piersicului nr. 1/3, bl. C, ap. 4
15	164	1	251695	Splaiul Mircea cel Bătrân nr. 8B, bl. 31, ap. 52

```

create table ANGAJAT (
  id_angajat number(5) constraint id_angajat_pk primary key,
  nume varchar2(30) constraint nume_nn not null,
  prenume varchar2(30) constraint prenume_nn not null,
  id_adresa number(5) constraint angajat_adresa_nn not null,
  nr_telefon varchar2(15) constraint nr_telefon_nn not null,
  email varchar2(320),
  cnp varchar(13) constraint cnp_nn not null,
  foreign key (id_adresa) references ADRESA(id_adresa)
);

create sequence angajati
start with 300
increment by 1;

insert into angajat(id_angajat,nume,prenume,id_adresa,nr_telefon,cnp)
values (angajati.nextval,'Ana','Macovei',151,'0273833014','2910509017336');
insert into angajat(id_angajat,nume,prenume,id_adresa,nr_telefon,cnp)
values (angajati.nextval,'Tudor','Mihai',153,'0771267946','1910273219266');
insert into angajat(id_angajat,nume,prenume,id_adresa,nr_telefon,cnp)
values (angajati.nextval,'Lidia','Ardelean',158,'0273833014','2970325016504');
insert into angajat
values (angajati.nextval,'Lazar','Niculescu',155,'0713702623','lazarcu@gmail.com','2910509017336');
insert into angajat
values (angajati.nextval,'Julia','Botezatu',150,'0363617583','juliabo@gmail.com','2970801416971');
insert into angajat

```

ID_ANGAJAT	NUME	PRENUME	ID_ADRESA	NR_TELEFON	EMAIL	CNP
1	300 Ana	Macovei	151	0273833014	(null)	2910509017336
2	301 Tudor	Mihai	153	0771267946	(null)	1910273219266
3	302 Lidia	Ardelean	158	0273833014	(null)	2970325016504
4	303 Lazar	Niculescu	155	0713702623	lazarcu@gmail.com	2910509017336
5	304 Julia	Botezatu	150	0363617583	juliabo@gmail.com	2970801416971
6	320 Nelu	Teodorescu	151	0351276737	nelutescu@gmail.com	1910305417705
7	340 Gina	Ene	153	0230865520	ginane@gmail.com	2880616314790
8	341 Avram	Mihai	154	0249576019	avramula@gmail.com	1990305103710
9	342 Marcel	Simionescu	156	0362000525	marcescu@gmail.com	1950416071974
10	343 Delia	Blaga	152	0756529565	deliablaga@gmail.com	2981128234029
11	344 Ortansa	Toma	151	0780347918	ortama@gmail.com	2900405098050
12	345 Laurentiu	Nicolae	161	0219039494	laurnic@gmail.com	1860712366795
13	346 Jan	Filip	163	0719319182	janflip@gmail.com	5020202209100
14	347 Valentina	Chirila	164	0746203124	valela@gmail.com	6010528100830
15	348 Timotei	Rusu	160	0244886450	timotsu@gmail.com	1871221321071


```

create table SEDIU(
  id_sediu number(5) constraint id_sediu_pk primary key,
  id_adresa number(5) constraint sediu_adresa_nn not null,
  foreign key (id_adresa) references ADRESA(id_adresa)
);

create sequence sedii
start with 500
increment by 1;

insert into sediu
values (sedii.nextval,152);
insert into sediu
values (sedii.nextval,154);
insert into sediu
values (sedii.nextval,156);
insert into sediu
values (sedii.nextval,157);
insert into sediu
values (sedii.nextval,159);

select *
from sediu;

```

Script Output x Query Result x

SQL | All Rows Fetched: 5 in 0.003 seconds

ID_SEDIU	ID_ADRESA
1	500 152
2	501 154
3	502 156
4	503 157
5	504 159

```

create table INSTRUCTOR (
  id_angajat number(5) constraint id_instructor_pk primary key,
  data_carnet date constraint data_carnet_nn not null,
  foreign key (id_angajat) references ANGAJAT(id_angajat)
);

insert into instructor
values (301,to_date('11/03/1998','DD/MM/YYYY'));
insert into instructor
values (303,to_date('27/05/2001','DD/MM/YYYY'));
insert into instructor
values (304,to_date('13/01/2004','DD/MM/YYYY'));
insert into instructor
values (343,to_date('13/01/2004','DD/MM/YYYY'));
insert into instructor
values (347,to_date('13/01/2004','DD/MM/YYYY'));

select *
from instructor;

```

Script Output x Query Result x

SQL | All Rows Fetched: 5 in 0.003 seconds

ID_ANGAJAT	DATA_CARNET
1	301 11-MAR-98
2	303 27-MAY-01
3	304 13-JAN-04
4	343 13-JAN-04
5	347 13-JAN-04

```

create table PROF_LEGIS(
  id_angajat number(5) constraint id_prof_pk primary key,
  tip_fac varchar2(25) constraint tip_fac_nn not null,
  data_exam date constraint data_exam_nn not null,
  constraint CHK_tip_fac check (lower(tip_fac) in ('tehnic','juridic')),
  foreign key (id_angajat) references ANGAJAT(id_angajat)
);

insert into prof_legis
values (302,'Tehnic',to_date('27/04/2019','DD/MM/YYYY'));
insert into prof_legis
values (320,'Tehnic',to_date('13/02/2016','DD/MM/YYYY'));
insert into prof_legis
values (340,'Juridic',to_date('05/07/2020','DD/MM/YYYY'));
insert into prof_legis
values (341,'Tehnic',to_date('12/07/2020','DD/MM/YYYY'));
insert into prof_legis
values (342,'Juridic',to_date('04/09/2018','DD/MM/YYYY'));

select *
from prof_legis;

```

```

create table SECRETAR(
  id_angajat number(5) constraint id_secretar_pk primary key,

```

Script Output x Query Result x

SQL | All Rows Fetched: 5 in 0.003 seconds

ID_ANGAJAT	TIP_FAC	DATA_EXAM
1	302 Tehnic	27-APR-19
2	320 Tehnic	13-FEB-16
3	340 Juridic	05-JUL-20
4	341 Tehnic	12-JUL-20
5	342 Juridic	04-SEP-18

```

create table SECRETAR(
  id_angajat number(5) constraint id_secretar_pk primary key,
  id_sediu number(5) constraint secretar_sediu_nn not null,
  foreign key (id_angajat) references ANGAJAT(id_angajat),
  foreign key (id_sediu) references SEDIU(id_sediu)
);

insert into secretar
values (300,501);
insert into secretar
values (344,502);
insert into secretar
values (345,503);
insert into secretar
values (346,504);
insert into secretar
values (348,500);

select *
from secretar;

```

```

create table AUTOMOBIL(

```

Script Output x Query Result x

SQL | All Rows Fetched: 5 in 0.004 seconds

ID_ANGAJAT	ID_SEDIU
1	300 501
2	344 502
3	345 503
4	346 504
5	348 500

```

create table AUTOMOBIL(
id_automobil number(5) constraint id_automobil_pk primary key,
marca varchar2(30) constraint marca_nn not null,
model varchar2(30),
categorie_automobil varchar2(2) constraint categorie_nn not null,
constraint CHK_cat_auto check(lower(categorie_automobil) in ('a1','a2','b1','be','b'))
);

create sequence automobile
start with 750
increment by 1;

insert into automobil
values (automobile.nextval,'Honda','CBR 125R','A1');
insert into automobil
values (automobile.nextval,'Ducati','Cafe Racer','A2');
insert into automobil
values (automobile.nextval,'Aixam','Coupe GTI','B1');
insert into automobil
values (automobile.nextval,'Volkswagen','Passat','B');
insert into automobil(id_automobil,marca,categorie_automobil)
values (automobile.nextval,'BMW','BE');

select *
from automobil;

```

Script Output x Query Result x

SQL | All Rows Fetched: 5 in 0.005 seconds

ID_AUTOMOBIL	MARCA	MODEL	CATEGORIE_AUTOMOBIL
1	750 Honda	CBR 125R	A1
2	751 Ducati	Cafe Racer	A2
3	752 Aixam	Coupe GTI	B1
4	753 Volkswagen	Passat	B
5	754 BMW	(null)	BE

```

create table CLIENT(
id_client number(5) constraint id_client_pk primary key,
nume varchar2(30) constraint nume_client_nn not null,
prenume varchar2(30) constraint prenume_client_nn not null,
data_nasterii date constraint datanasterii_nn not null,
id_adresa number(5) constraint client_adresa_nn not null,
nr_telefon varchar2(15) constraint nr_telefon_client_nn not null,
email varchar2(320),
cnp varchar(13) constraint cnp_client_nn not null,
foreign key (id_adresa) references ADRESA(id_adresa)
);

create sequence clienti
start with 1000
increment by 1;

insert into client
values (clienti.nextval,'Anuta','Stoica',to_date('06/12/2000','DD/MM/YYYY'),164,'0718465446','anutaca@gmail.com','2880904356822');
insert into client
values (clienti.nextval,'Visaron','Ionescu',to_date('02/03/1989','DD/MM/YYYY'),162,'0781572802','visaion@gmail.com','1860622454782');
insert into client
values (clienti.nextval,'Lavinia','Farcas',to_date('13/08/1999','DD/MM/YYYY'),161,'0753974373','lavinias@gmail.com','2921116016491');
insert into client
values (clienti.nextval,'Calin','Catana',to_date('25/03/2001','DD/MM/YYYY'),163,'0377556929','calincana@gmail.com','1880915165075');
insert into client
values (clienti.nextval,'Isabela','Pana',to_date('16/04/1992','DD/MM/YYYY'),158,'0346918264','isabelana@gmail.com','2960331113593');

select *
from client;

create table LECTIE_PRACTICA(
id_lectiepr number(5) constraint id_lectiepr_pk primary key,
id_instructor number(5) constraint lpr_instructor_nn not null,
id_elev number(5) constraint lpr_elev_nn not null,
id_automobil number(5) constraint lpr_automobil_nn not null,
data date constraint data_lpr_nn not null,
km_narcursi float(5) constraint km_nn not null
);

```

Script Output x Query Result x

SQL | All Rows Fetched: 5 in 0.002 seconds

ID_CLIENT	NUME	PRENUME	DATA_NASTERII	ID_ADRESA	NR_TELEFON	EMAIL	CNP
1	1000 Anuta	Stoica	06-DEC-00	164	0718465446	anutaca@gmail.com	2880904356822
2	1001 Visaron	Ionescu	02-MAR-89	162	0781572802	visaion@gmail.com	1860622454782
3	1002 Lavinia	Farcas	13-AUG-99	161	0753974373	lavinias@gmail.com	2921116016491
4	1003 Calin	Catana	25-MAR-01	163	0377556929	calincana@gmail.com	1880915165075
5	1004 Isabela	Pana	16-APR-92	158	0346918264	isabelana@gmail.com	2960331113593

```

create table LECTIE_PRACTICA(
  id_lectiepr number(5) constraint id_lectiepr_pk primary key,
  id_instructor number(5) constraint lpr_instructor_nn not null,
  id_elev number(5) constraint lpr_elev_nn not null,
  id_automobil number(5) constraint lpr_automobil_nn not null,
  data date constraint data_lpr_nn not null,
  km_parcursi float(5) constraint km_nn not null,
  foreign key (id_instructor) references INSTRUCTOR(id_angajat),
  foreign key (id_elev) references CLIENT(id_client),
  foreign key (id_automobil) references AUTOMOBIL(id_automobil)
);

create sequence practice
start with 1500
increment by 1;

insert into lectie_practica
values (practice.nextval,303,1001,750,to_date('23/07/2020','DD/MM/YYYY'),23);
insert into lectie_practica
values (practice.nextval,303,1002,750,to_date('25/07/2020','DD/MM/YYYY'),20);
insert into lectie_practica
values (practice.nextval,301,1001,752,to_date('23/07/2020','DD/MM/YYYY'),21.5);
insert into lectie_practica
values (practice.nextval,343,1003,753,to_date('28/07/2020','DD/MM/YYYY'),24.8);
insert into lectie_practica
values (practice.nextval,347,1004,754,to_date('02/08/2020','DD/MM/YYYY'),19.5);

select *
from lectie_practica;

```

Script Output x Query Result x

SQL | All Rows Fetched: 5 in 0.006 seconds

ID_LECTIEPR	ID_INSTRUCTOR	ID_ELEV	ID_AUTOMOBIL	DATA	KM_PARCURSI
1	1500	303	1001	750 23-JUL-20	23
2	1501	303	1002	750 25-JUL-20	20
3	1502	301	1001	752 23-JUL-20	22
4	1503	343	1003	753 28-JUL-20	25
5	1504	347	1004	754 02-AUG-20	20

```

create table contract(
  id_contract number(5) constraint id_contract_pk primary key,
  id_secretar number(5) constraint contract_secretar_nn not null,
  id_client number(5) constraint contract_client_nn not null,
  data_semnare date constraint contract_data_nn not null,
  detalii_contract varchar2(255) constraint detalii_nn not null,
  foreign key (id_secretar) references SECRETAR(id_angajat),
  foreign key (id_client) references CLIENT(id_client)
);

create sequence contracte
start with 2000
increment by 1;

insert into contract
values (contracte.nextval,344,1000,to_date('13/3/2020','DD/MM/YYYY'),'Pret: 2200 lei');
insert into contract
values (contracte.nextval,346,1001,to_date('17/07/2020','DD/MM/YYYY'),'Pret: 2450 lei');
insert into contract
values (contracte.nextval,348,1002,to_date('19/07/2020','DD/MM/YYYY'),'Pret: 2200 lei');
insert into contract
values (contracte.nextval,346,1003,to_date('20/07/2020','DD/MM/YYYY'),'Pret: 1400 lei, Categoria A1');
insert into contract
values (contracte.nextval,300,1004,to_date('28/09/2020','DD/MM/YYYY'),'Pret 2200 lei');

select *
from contract;

```

Script Output x Query Result x

SQL | All Rows Fetched: 5 in 0.006 seconds

ID_CONTRACT	ID_SECRETAR	ID_CLIENT	DATA_SEMNARE	DETALII_CONTRACT
1	2001	344	1000 13-MAR-20	Pret: 2200 lei
2	2002	346	1001 17-JUL-20	Pret: 2450 lei
3	2003	348	1002 19-JUL-20	Pret: 2200 lei
4	2004	346	1003 20-JUL-20	Pret: 1400 lei, Categoria A1
5	2005	300	1004 28-SEP-20	Pret 2200 lei

```

create table plata(
id_plata number(5) constraint plata_pk primary key,
id_contract number(5) constraint plata_contract_nn not null,
data date,
suma number(10) constraint suma_nn not null,
foreign key (id_contract) references CONTRACT(id_contract)
);

create sequence plati
start with 2500
increment by 1;

insert into plata
values (plati.nextval,2001,to_date('17/03/2020','DD/MM/YYYY'),750);
insert into plata
values (plati.nextval,2002,to_date('18/08/2020','DD/MM/YYYY'),500);
insert into plata
values (plati.nextval,2003,to_date('19/06/2020','DD/MM/YYYY'),1000);
insert into plata
values (plati.nextval,2003,to_date('27/06/2020','DD/MM/YYYY'),1000);
insert into plata
values (plati.nextval,2003,to_date('05/08/2020','DD/MM/YYYY'),200);

select *
from plata;

```

Script Output x Query Result x

SQL | All Rows Fetched: 5 in 0.005 seconds

ID_PLATA	ID_CONTRACT	DATA	SUMA
1	2500	2001 17-MAR-20	750
2	2501	2002 18-AUG-20	500
3	2502	2003 19-JUN-20	1000
4	2503	2003 27-JUN-20	1000
5	2504	2003 05-AUG-20	200

```

create table LECTIE_TEORETICA(
id_lectietr number(5) constraint lectietr_pk primary key,
id_profesor number(5) constraint teor_prof_nn not null,
id_sediu number(5) constraint teor_sediu_nn not null,
data date constraint teor_data_nn not null,
detalii_suplimentare varchar2(150),
foreign key (id_profesor) references PROF_LEGIS(id_angajat),
foreign key (id_sediu) references SEDIU(id_sediu)
);

create sequence teoretice
start with 3000
increment by 1;

insert into lectie_teoretica
values (teoretice.nextval,342,500,to_date('25/03/2021','DD/MM/YYYY'),'Chestionare');
insert into lectie_teoretica
values (teoretice.nextval,320,501,to_date('16/04/2021','DD/MM/YYYY'),'Intrebari indicatoare');
insert into lectie_teoretica
values (teoretice.nextval,320,502,to_date('23/02/2021','DD/MM/YYYY'),'Intrebari semnale');
insert into lectie_teoretica
values (teoretice.nextval,340,501,to_date('15/01/2021','DD/MM/YYYY'),'Intrebari mecanica');
insert into lectie_teoretica
values (teoretice.nextval,341,504,to_date('10/02/2021','DD/MM/YYYY'),'Chestionare');

select *
from lectie_teoretica;

```

Script Output x Query Result x

SQL | All Rows Fetched: 5 in 0.006 seconds

ID_LLECTIETR	ID_PROFESOR	ID_SEDIU	DATA	DETALII_SUPLIMENTARE
1	3000	342	500 25-MAR-21	Chestionare
2	3001	320	501 16-APR-21	Intrebari indicatoare
3	3002	320	502 23-FEB-21	Intrebari semnale
4	3003	340	501 15-JAN-21	Intrebari mecanica
5	3004	341	504 10-FEB-21	Chestionare

```

create table PRESENTA(
  id_lectie number(5) constraint presenta_lectie_nn not null,
  id_elev number(5) constraint presenta_elev_nn not null,
  foreign key (id_lectie) references lectie_teoretica(id_lectietr),
  foreign key (id_elev) references client(id_client)
);

insert into presenta
values (3000,1002);
insert into presenta
values (3000,1003);
insert into presenta
values (3002,1002);
insert into presenta
values (3003,1002);
insert into presenta
values (3003,1003);
insert into presenta
values (3001,1000);
insert into presenta
values (3004,1001);
insert into presenta
values (3004,1003);
insert into presenta
values (3003,1001);
insert into presenta
values (3002,1000);

select *
from presenta;

```

ID_LECTIE	ID_ELEV
1	3000 1002
2	3000 1003
3	3003 1002
4	3003 1003
5	3001 1000
6	3004 1001
7	3004 1003
8	3003 1001
9	3002 1000

```

create table EXAMEN(
  id_examen number(5) constraint examen_id_pk primary key,
  id_elev number(5) constraint examen_elev_nn not null,
  data date constraint examen_data_nn not null,
  calificativ varchar2(10) constraint examen_cal_nn not null,
  foreign key (id_elev) references CLIENT(id_client),
  constraint CHK_CAL CHECK (lower(calificativ) in ('admis','respins'))
);

create sequence examene
start with 3500
increment by 1;

insert into examen
values (examene.nextval,1000,to_date('27/11/2019','DD/MM/YYYY'),'Admis');
insert into examen
values (examene.nextval,1000,to_date('13/01/2020','DD/MM/YYYY'),'Admis');
insert into examen
values (examene.nextval,1001,to_date('13/05/2020','DD/MM/YYYY'),'Respins');
insert into examen
values (examene.nextval,1001,to_date('25/07/2020','DD/MM/YYYY'),'Respins');
insert into examen
values (examene.nextval,1001,to_date('27/08/2020','DD/MM/YYYY'),'Admis');
insert into examen
values (examene.nextval,1002,to_date('14/05/2020','DD/MM/YYYY'),'Respins');
insert into examen
values (examene.nextval,1002,to_date('27/05/2020','DD/MM/YYYY'),'Respins');
insert into examen
values (examene.nextval,1003,to_date('10/07/2020','DD/MM/YYYY'),'Admis');
insert into examen
values (examene.nextval,1003,to_date('11/07/2020','DD/MM/YYYY'),'Respins');
insert into examen
values (examene.nextval,1003,to_date('23/09/2020','DD/MM/YYYY'),'Admis');

```

ID_EXAMEN	ID_ELEV	DATA	CALIFICATIV
1	3500	1000 27-NOV-19	Admis
2	3501	1000 13-JAN-20	Admis
3	3502	1001 13-MAY-20	Respins
4	3503	1001 25-JUL-20	Respins
5	3504	1001 27-AUG-20	Admis
6	3505	1002 14-MAY-20	Respins

```

create table TEORETIC(
  id_examen number(5) constraint examen_teoretic_pk primary key,
  punctaj number(2) constraint teoretic_punct_nn not null,
  foreign key (id_examen) references examen(id_examen)
);

insert into teoretic
values (3500,23);
insert into teoretic
values (3502,19);
insert into teoretic
values (3503,20);
insert into teoretic
values (3504,22);
insert into teoretic
values (3505,16);

select *
from teoretic;

```

ID_EXAMEN	PUNCTAJ
1	3500
2	3502
3	3503
4	3504
5	3505

```

create table PRACTIC(
  id_examen number(5) constraint examen_practic_pk primary key,
  puncte_penalizare number(3) constraint pct_nn not null,
  foreign key (id_examen) references examen(id_examen)
);

insert into practic
values (3501,5);
insert into practic
values (3506,36);
insert into practic
values (3507,19);
insert into practic
values (3508,102);
insert into practic
values (3509,0);

select *
from practic;

```

ID_EXAMEN	PUNCTE_PENALIZARE
1	3501
2	3506
3	3507
4	3508
5	3509

Cereri SQL

```
-- Punctul 11

-- Sa se afiseze nume si prenumele elevului, instructorului si marca --
-- de masina tuturor lectiilor practice ce au avut loc dupa 24 Iulie --
-- 2020 --

select c.nume,c.prename,aa.nume,aa.prename, a.marca, l.data
from client c,lectie_practica l,automobil a,instructor i, angajat aa
where (c.id_client = l.id_elev)
and (a.id_automobil = l.id_automobil)
and (i.id_angajat = l.id_instructor)
and (i.id_angajat = aa.id_angajat)
and (l.data > to_date('24/07/2020','DD/MM/YYYY'))
order by 6;
```

Script Output x Query Result x

SQL | All Rows Fetched: 3 in 0.013 seconds

	NUME	PRENUME	NUME_1	PRENUME_1	MARCA	DATA
1	Lavinia	Farcas	Lazar	Niculescu	Honda	25-JUL-20
2	Calin	Catana	Delia	Blaga	Volkswagen	28-JUL-20
3	Isabela	Pana	Valentina	Chirila	BMW	02-AUG-20

```
-- Sa se afiseze numele, prenumele si numarul de examene date al --
-- fiecarui client --

select nume,prename, (select count(*)
                        from examen e
                        where e.id_elev = c.id_client) "Nr. Examene"
from client c

-- Sa se afiseze numele, prenumele, adresa, emailul si --
```

Script Output x Query Result x

SQL | All Rows Fetched: 5 in 0.01 seconds

	NUME	PRENUME	Nr. Examene
1	Anuta	Stoica	2
2	Visaron	Ionescu	3
3	Lavinia	Farcas	2
4	Calin	Catana	3
5	Isabela	Pana	0


```

-- Sa se afiseze numele, prenumele, adresa, emailul si
-- codul sediiului tuturor angajatilor care lucreaza ca secretar
-- si celor carora nu le este setat email-ul, sa se afis in schimb
-- mesajul Mail Neseatat

select a.numa,a.prenume,o.denumire,aa.info_adresa,s.id_sediu, nvl(a.email,'Mail Neseatat')
from angajat a,secretar s,adresa aa,oras o
where (a.id_angajat = s.id_angajat)
and (a.id_adresa = aa.id_adresa)
and (aa.id_oras = o.id_oras);

```

Script Output x Query Result x

SQL | All Rows Fetched: 5 in 0.01 seconds

	NUME	PRENUME	DENUMIRE	INFO_ADRESA	ID_SEDIU	NVL(A.EMAIL,'MAILNESETAT')
1	Jan	Filip	Bucuresti	Splaiul Piersicului nr. 1/3, bl. C, ap. 4	504	janflip@gmail.com
2	Ana	Macovei	Focsani	Str. Stadionului, Nr. 45, Bl.2	501	Mail Neseatat
3	Ortansa	Toma	Focsani	Str. Stadionului, Nr. 45, Bl.2	502	ortama@gmail.com
4	Laurentiu	Nicolae	Focsani	Str. Vlad Tepeş 3, Mun. Piatra-Olt, Alba	503	laurnic@gmail.com
5	Timotei	Rusu	Buzau	Splaiul Mircea cel Bătrân nr. 24, bl. B, ap. 96	500	timotsu@gmail.com

```

-- Sa se afiseze numele, prenumele si rangul tuturor instructorilor -
-- Rangul se calculeaza astfel; <10 ani cu carnet => Inceptor
-- <20 ani cu carnet => Intermediar
-- >=20 ani cu carnet => Avansat

select a.numa,a.prenume,
CASE
WHEN 2021-extract(year from i.data_carnet)<10 THEN 'Inceptor'
WHEN 2021-extract(year from i.data_carnet)<20 THEN 'Intermediar'
ELSE 'Avansat'
END AS Rang
from angajat a,instructor i
where (a.id_angajat = i.id_angajat);

```

Script Output x Query Result x

SQL | All Rows Fetched: 5 in 0.005 seconds

	NUME	PRENUME	RANG
1	Tudor	Mihai	Avansat
2	Lazar	Niculescu	Avansat
3	Julia	Botezatu	Intermediar
4	Delia	Blaga	Intermediar
5	Valentina	Chirila	Intermediar

```
-- Sa se afiseze angajatii ce au un nume mai mic de 4 litere sau --
-- care au prenumele incepand cu litera T --
```

```
select nume,prenume
from angajat a
where (length(a.nume)<4)
or (lower(a.prenume) like 't%');
```

Script Output x Query Result x

SQL | All Rows Fetched: 4 in 0.005 seconds

	NUME	PRENUME
1	Ana	Macovei
2	Nelu	Teodorescu
3	Ortansa	Toma
4	Jan	Filip

```
-- Sa se afiseze id-urile sediilor precum si codurile lor interne --
```

```
select sediu.id_sediu, decode(sediu.id_sediu,500,'Alpha'
,501,'Beta'
,502,'Charlie'
,503,'Delta'
,504,'Foxtrot') Cod
from sediu;
```

Script Output x Query Result x

SQL | All Rows Fetched: 5 in 0.006 seconds

	ID_SEDIU	COD
1	500	Alpha
2	501	Beta
3	502	Charlie
4	503	Delta
5	504	Foxtrot

```
-- Sa se afiseze totii instructorii care au utilizat automobilul cu--
-- id-ul 750 la orele de pregatire practica --
```

```
select angajat.ume,angajat.prenume
from angajat
where (angajat.id_angajat in (select id_angajat
                             from instructor
                             where (id_angajat in (select id_instructor
                                                    from lectie_practica
                                                    where (id_automobil = 750)))));
```

Script Output x Query Result x

SQL | All Rows Fetched: 1 in 0.008 seconds

	NUME	PRENUME
1	Lazar	Niculescu

```
-- Sa se afiseze id-ul tuturor lectiilor practice la care au --
-- participat clienti ce au semnat un contract cu secretarul cu --
-- id-ul 346 --
```

```
select 1.id_lectiepr
from lectie_practica 1
where (1.id_elev = (select id_client
                   from client c
                   where (c.id_client = 1.id_elev)
                   and (c.id_client = (select cc.id_client
                                       from contract cc
                                       where (cc.id_secretar = 346)
                                       and (cc.id_client = c.id_client))));
```

Script Output x Query Result x

SQL | All Rows Fetched: 3 in 0.007 seconds

	ID_LECTIEPR
1	1500
2	1502
3	1503

```
-- Sa se afiseze (daca si numai daca exista cel putin 4 astfel de examene)
-- numarul de examene ce au calificativul respins si data celui mai recent
-- cat si numarul de examene cu calificativul admis si data celui mai recent

select calificativ,max(data),count(*)
from examen
group by calificativ
having count(*)>4;
```

Script Output x Query Result x

SQL | All Rows Fetched: 2 in 0.005 seconds

	CALIFICATIV	MAX(DATA)	COUNT(*)
1	Admis	23-SEP-20	5
2	Respins	25-JUL-20	5

<pre> -- Sa se afiseze numarul maxim de lectii practice existente in aceeasi zi </pre>					
<pre> select max(count(*)) from lectie_practica group by data; </pre>					
<div>Script Output x Query Result x</div> <div>SQL All Rows Fetched: 1 in 0.004 seconds</div> <table> <tr> <th></th><th>MAX(COUNT(*))</th></tr> <tr> <td>1</td><td>2</td></tr> </table>			MAX(COUNT(*))	1	2
	MAX(COUNT(*))				
1	2				

</

```

-- Sa se afiseze totii instructorii care au utilizat automobile cu --
-- marca Honda la orele de pregatire practica cu elevul cu numele --
-- 'Lavinia' --
-----
select angajat.num,angajat.preume
from angajat
where (angajat.id_angajat in (select id_instructor
                             from lectie_practica l, client c, automobil a
                             where (lower(a.marca) = 'honda')
                             and (lower(c.num) = 'lavinia')
                             and (c.id_client = l.id_elev)
                             and (a.id_automobil = l.id_automobil)));

```

Query Result x

SQL | All Rows Fetched: 1 in 0.012 seconds

	NUME	PRENUME
1	Lazar	Niculescu

```

-- Sa se afiseze id-ul secretarului care a realizat contractul la --
-- care a fost realizata plata cu id-ul 2502 si care a fost semnat --
-- de clientul cu id-ul 1002 --
-----
select id_angajat
from secretar s
where (s.id_angajat = (select c.id_secretar
                      from contract c, client cc, plata p
                      where (c.id_client = cc.id_client)
                      and (c.id_secretar = s.id_angajat)
                      and (p.id_contract = c.id_contract)
                      and (cc.id_client = 1002)
                      and (p.id_plata = 2502)
                      ));

```

Query Result x

SQL | All Rows Fetched: 1 in 0.003 seconds

	ID_ANGAJAT
1	348

Operatii Suprimare / Actualizare

```
-- Punctul 12

rollback;

update angajat
set email = null      -- Stergerea emailurilor tuturor angajatiilor
where (id_angajat in (select id_angajat
                      from prof_legis)) -- care lucreaza ca profesori de legislatie
```

Script Output x Query Result x

Task completed in 0.04 seconds

5 rows updated.

```
update client -- Schimbarea numelui clientilor
set nume = 'Ion' -- in Ion tuturor clientilor care
where prenume like '%r%' -- au litera 'r' in prenume
and (id_adresa in (select id_adresa -- si care locuiesc la o adresa din
                  from adresa -- orasul cu id-ul 4
                  where (id_oras = 4)))
```

Script Output x Query Result x

Task completed in 0.041 seconds

1 row updated.

```
-----
update secretar -- Schimbarea sediului unui secretar
set id_sediu = &id -- intr-un sediu ales de la tastatura
where (id_angajat in (select distinct id_secretar
                      from contract)) -- la secretarii ce au facut
                                     -- macar un contract
-----
```

Script Output x Query Result x

Task completed in 3.024 seconds

```
old:update secretar -- Schimbarea sediului unui secretar
set id_sediu = &id -- intr-un sediu ales de la tastatura
where (id_angajat in (select distinct id_secretar
                      from contract)) -- la secretarii ce au facut
                                     -- macar un contract

new:update secretar -- Schimbarea sediului unui secretar
set id_sediu = 500 -- intr-un sediu ales de la tastatura
where (id_angajat in (select distinct id_secretar
                      from contract)) -- la secretarii ce au facut
                                     -- macar un contract

4 rows updated.
```

```
-----
insert into oras
values (25,'Botosani');

delete -- Se sterg toate orasele care nu au vreo adresa la ele
from oras
where (id_oras not in (select id_oras from adresa));
-----
```

Script Output x Query Result x

Task completed in 0.029 seconds

```
0 rows deleted.

1 row inserted.

1 row deleted.
```

```
delete
from automobil -- Se sterg din sistem masinile neutilizate la orele de pregatire practica
where (marca not in (select distinct marca from automobil a,lectie_practica l where a.id_automobil=l.id_automobil))
-- Este sters 751
```

Script Output x Query Result x

Task completed in 0.04 seconds

1 row deleted.

```
delete
from plata
where (id_contract in (select id_contract -- Stergerea platilor facut de clientul cu id-ul
                        from contract      -- ales de la tastatura
                        where (id_client = &idc)))
```

Script Output x Query Result x

Task completed in 1.272 seconds

old:delete
from plata
where (id_contract in (select id_contract -- Stergerea platilor facut de clientul cu id-ul
 from contract -- ales de la tastatura
 where (id_client = &idc)))

new:delete
from plata
where (id_contract in (select id_contract -- Stergerea platilor facut de clientul cu id-ul
 from contract -- ales de la tastatura
 where (id_client = 1002)))

3 rows deleted.