

**PROIECT BAZE DE DATE**

Proiectarea unei baze de date privind evidența închirierii de  
motociclete

Student: Ion Adina

An universitar: 2022-2023

Grupa: 212

Seria: 21

## **Cuprins**

1. Descrierea modelului real ,a utilității acestuiași a regulilor de funcționare.
  2. Prezentarea constrângerilor (restricții, reguli) impuse asupra modelului.
  3. Descrierea entităților, incluzând precizarea cheii primare.
  4. Descrierea relațiilor, incluzând precizarea cardinalității acestora.
  5. Descrierea atributelor, incluzând tipul de date și eventualele constrângeri, valori implicite, valori posibile ale atributelor.
  6. Realizarea diagramei entitate-relație corespunzătoare descrierii de la punctele 3-5.
  7. Realizarea diagramei conceptuale corespunzătoare diagramei entitate-relație proiectate la punctul 6.Diagrama conceptuală obținută trebuie să conțină minimum 6 tabele (fără considerarea subentităților), dintre care cel puțin un tabel asociativ.
  8. Enumerarea schemelor relaționale corespunzătoare diagramei conceptual eproiectate la punctul 7.
  9. Realizarea normalizării până la forma normală 3(FN1-FN3).
  10. Crearea unei secvențe ce va fi utilizată în inserarea înregistrărilor în tabele (punctul 11).
  11. Crearea tabelelor în SQL și inserarea de date coerente în fiecare dintre acestea (minimum 5 înregistrări înfiecare tabel neasociativ; minimum10 înregistrări în tabelele asociative).
  12. Formulați în limbaj natural și implementați 5cereri SQL complexe ce vor utiliza, în ansamblul lor, următoarele elemente:
    - subcereri sincronizate în care intervin cel puțin 3 tabele
    - subcereri nesincronizate în clauza FROM
    - grupări de date cu subcereri nesincronizate in care intervin cel puțin 3 tabele, funcții grup, filtrare la nivel de grupuri(in cadrul aceleiasi cereri)
    - ordonări si utilizarea funcțiilor NVL și DECODE (in cadrul aceleiasi cereri)
    - utilizarea a cel puțin 2 funcții pe șiruri de caractere, 2 funcții pe date calendaristice, a cel puțin unei expresii CASE
    - utilizarea a cel puțin 1 bloc de cerere(clauza WITH)
- Observație:**Într-o cerere se vor regăsi mai multe elemente dintre cele enumerate mai sus, astfel încât cele 5 cereri să le cuprindă pe toate.

- 13.** Implementarea a 3 operații de actualizare și de suprimare a datelor utilizând subcereri.
- 14.** Crearea unei vizualizări complexe. Dați un exemplu de operație LMD permisă pe vizualizarea respectivă și un exemplu de operație LMD nepermisă.
- 15.** Formulați în limbaj natural și implementați în SQL: o cerere ce utilizează operația outer-join pe minimum 4 tabele, o cerere ce utilizează operația division și o cerere care implementează analiza top-n. Observație: Cele 3 cereri sunt diferite de cererile de la exercițiul 12.
- 16.** Optimizarea unei cereri, aplicând regulile de optimizare ce derivă din proprietățile operatorilor algebrei relaționale. Cererea va fi exprimată prin expresie algebrică, arbore algebric și limbaj (SQL), atât anterior cât și ulterior optimizării.
- 17.**
- a. Realizarea normalizării BCNF, FN4, FN5
  - b. Aplicarea denormalizării, justificând necesitatea acesteia.

**1. Descrierea modelului real, a utilității acestuia și a regulilor de funcționare.**

Motocicleta reprezintă un mijloc de transport rapid care are multe avantaje: investiția se amortizează mai repede; fiabilitatea este mai mare față de un autoturism; doza de adrenalină este mult mai mare pe motor și călătoriile sunt mai antrenante; vizibilitate mai mare și peisajele pot fi admirate mult mai bine.

Motivația alegerii acestei teme, este strâns legată de pasiunea mea pentru motociclete, dorind să fac o legătură între pasiune și realizarea proiectului, astfel că baza mea de date gestionează închirieri de motociclete.

Proiectul propune o bază de date care să permită gestiunea motocicletelor dintr-o flotă ce dorește să le închirieze. Acestea au nevoie de reparații, pentru care se percepe și un comision, sau de întrețineri ce se efectuează periodic. Clienții închiriază și pentru orice închiriere există și o plată, care se poate efectua la o altă dată față de cea la care se finalizează închirierea (clientul poate avea un anumit termen de plată). Motocicletele sunt descrise video, prin text și imagini.

**2. Prezentarea constrângerilor (restricții, reguli) impuse asupra modelului.**

- ✓ O anumită motocicletă provine de la un singur distribuitor. Nu pot exista două produse la fel dar de la distribuitori diferiți.
- ✓ Numerele de telefon salvate în baza de date nu vor conține prefixele țărilor.
- ✓ Adresele de email trebuie să fie valide (ex: adinaion53@gmail.com)

- ✓ Numele trebuie să conțină doar litere.

### 3.Descrierea entităților, incluzând precizarea cheii primare.

- Entitatea “Motocicleta” contine detalii despre motocicletele companiei,precum numar inmatriculare,model,marca,anul fabricarii si pretul inchirierii; cheia primara este reprezentata de *numar\_inmatriculare*.
- Entitatea “Inchiriere” contine detalii despre inchirierea de motociclete ,precum clientul care face inchirierea,numarul de inmatriculare al motocicletei inchiriate,data de inceput si data de sfarsit,pretul total; cheia primara este reprezentata de *id\_inchiriere* .
- Entitatea “Plata” contine detalii despre plata,precum clientul care face plata,id ul inchirierii ,pretul,data in care se face plata; cheia primara este reprezentata de *id\_plata*.
- Entitatea “Client” contine detalii despre clientul care face inchirierea,precum nume,prenume,adresa,email,telefon; cheia primara este reprezentata de *id\_client*.
- Entitatea “Intretinere” contine detalii despre intretinerea motocicletei,precum numarul de inmatriculare al motocicletei respective,data,tipul de intretinere si costurile implicate; cheia primara este reprezentata de *id\_intretinere*.
- Entitatea “Comision” contine detalii despre comision precum,id ul reparatiei si cat la suta comision se aplica; cheia primara este reprezentata de *id\_comision*.
- Entitatea “Reparatie” contine detalii despre reparatie ,precum numarul de inmatriculare al motocicletei respective,data la care se incepe reparatia, si costurile; cheia primara este reprezentata de *id\_reparatie*.
- Entitatea “Distribuitor” contine detalii despre cei care vand produsul companiei, urmarindu-se numele si prenumele distribuitorului,numarul de telefon, email-ul si orasul din care sunt vandute; cheia primara este reprezentata de *id\_distribuitor*.
- Entitatea “Imagine” contine detalii despre cum este promovat motocicleta din imagine,urmarindu-se numele imaginii,formatul,rezolutia,id ul motocicletei pentru care se realizeaza si link-ul catre imagine; cheia primara este reprezentata de *id\_imagine*.
- Entitatea “Video” contine detalii despre cum este descris produsul din videoclip,urmarindu-se numele videoclipului,durata,formatul,id-ul motocicletei pentru care se realizeaza si link-ul catre videoclip; cheia primara este reprezentata de *id\_video*.

- Entitatea “Motociclete\_distribuite” conține detalii despre motocicletele provenite de la distribuitori, urmarindu-se numarul de înmatriculare, id-ul distribuitorului și observatii despre ea.
- Entitatea “Motociclete\_inchiriate” contine detalii despre motocicletele închiriate de clienți, urmarindu-se numarul de înmatriculare, id-ul clientului și observații.
- Entitatea “Descriere” conține descrierea motocicletei, urmarindu-se formatul, descrierea și id-ul motocicletei; cheia primară este reprezentată de *id\_descriere*.
- Entitatea “Recenzie” conține parerea clienților despre motocicletele închiriate, urmarindu-se id ul motocicletei ,id-ul clientului si părerea pe care acesta o are despre produs; cheia primară este reprezentată prin *id\_recenzie*.
- Entitatea “Subcategorii” conține detalii despre subcategoria motocicletei, urmarindu-se numele subcategoriei. Cheia primară e reprezentată de *Id\_Subcategorie*.
- Entitatea “Categorii” conține detalii despre categoria motocicletei, urmarindu-se numele categoriei. Cheia primară e reprezentată de *Id\_categorie*.

#### 4. Descrierea relațiilor, incluzând precizarea cardinalității acestora.

Mai mulți distribuitori vând mai multe motociclete către o companie, motocicleta având un distribuitor (cardinalitate many to one), iar clienții pot face mai multe închirieri, o motocicleta avand un singur client la un moment respectiv (cardinalitate many to one), însa mai multe închirieri intr-un anumit interval de timp (many to many). Produsele pot avea recenzii primite de la clienti, un client poate lăsa mai multe recenzii, recenzia avand un client (cardinalitate many to one). O motocicleta este descrisă prin una sau mai multe imagini, video-uri, descrieri (cardinalitate one to many) . O motocicleta poate avea mai multe reparații și are sigur mai multe intretineri (cardinalitate one to many).

#### 5. Descrierea atributelor, incluzând tipul de date și eventualele constrângeri, valori implicite, valori posibile ale atributelor.

- Motocicleta
  - Numar\_inmatriculare: este identificatorul unic pentru fiecare motocicleta în parte. Va fi primary key. (CHAR(7))
  - Marca: va fi NOT NULL, conține numele mărcii (VARCHAR(255))

- An\_fabricatie: va fi NOT NULL întrucat continele anul fabricării motocicletei (INT)
- Pret\_inchiriere : va fi NOT NULL deoarece fiecare produs trebuie să aibă un preț de închiriere, iar acesta va fi întotdeauna pozitiv printr-o constrangere ce verifică că prețul să fie mai mare decat 0 (DECIMAL(10,2))

#### ➤ Inchiriere

- Id\_inchiriere: este identificatorul unic pentru fiecare închiriere în parte. Va fi primary key. (INTEGER)
- Id\_client : Asigură relația cu entitatea "Client".( INTEGER)
- Numar\_inmatriculare : Asigură relația cu entitatea "Motocicleta"( CHAR(7)).
- Data\_inceput : va fi NOT NULL deoarece reprezintă data cu care începe inchierea (DATE)
- Data\_sfarsit: va fi NOT NULL deoarece reprezintă data la care se incheie închirierea (DATE)
- Pret\_total :va fi NOT NULL deoarece fiecare produs trebuie să aiba un preț de inchiriere, iar acesta va fi întotdeauna pozitiv printr-o constrangere ce verifică că prețul să fie mai mare decat 0 (DECIMAL(10,2))

#### ➤ Plata

- Id\_plata :este identificatorul unic pentru fiecare plată în parte. Va fi primary key. (INTEGER)
- Id\_client: Asigură relația cu entitatea "Client".( INTEGER)
- Id\_inchiriere: Asigură relația cu entitatea "Inchiriere". ( INTEGER)
- Data\_plata (DATE)
- Pret: va fi NOT NULL deoarece fiecare produs trebuie să aibă o tranzacție pentru închiriere, iar acesta va fi întotdeauna pozitiv printr-o constrângere ce verifică că prețul să fie mai mare decat 0 (DECIMAL(8,2))

#### ➤ Client

- Id\_client: este identificatorul unic pentru fiecare client in parte. Va fi primary key. (INT)
- Nume: va fi NOT NULL, întrucat va conține numele de familie al clientului. Am adăugat o constrângere ce verifică ca numele să fie format doar din litere (VARCHAR(50))
- Prenume: va fi NOT NULL, întrucat va conține prenumele clientului. Am adăugat o constrângere ce verifică ca numele să fie format doar din litere (VARCHAR(50))

- Email: va fi NOT NULL reprezentând metoda de contact de rezervă a clientului și totodată UNIQUE deoarece email-ul este diferit de la individ la individ. Am mai creat o constrângere ce verifică folosind regex formatul mailului pentru a fi unul valid (VARCHAR(50))
- Telefon: va fi NOT NULL pentru a salva una din metodele de contact ale clientului și totodată UNIQUE deoarece numerele de telefon inserate sunt românești, deși nu mai adaug prefix acestea vor fi unice pentru fiecare persoană în parte. Pe lângă aceasta am creat o constrângere care să verifice ca numărul de telefon să conțină doar cifre și încă una ce verifică că numărul de telefon conține 10 cifre acesta fiind numărul de cifre pe care le au numerele de telefon românești(VARCHAR(50))

#### ➤ Intretinere

- Id\_intretinere: este identificatorul unic pentru fiecare întreținere în parte. Va fi primary key. (INTEGER)
- Numar\_inmatriculare: Asigură relația cu entitatea “Motocicleta”( CHAR(7)).
- Data: va fi NOT NULL deoarece reprezintă data la care se efectuează întreținerea (DATE)
- Cost: va fi NOT NULL deoarece fiecare motocicletă are costuri pentru întreținere (DECIMAL(10,2)).

#### ➤ Comision

- Id\_comision: este identificatorul unic pentru fiecare comision în parte. Va fi primary key. (INTEGER)
- Id\_reparatie: Asigură relația cu entitatea “Reparatie”( INTEGER).
- Procent\_comision : va fi NOT NULL, întrucât fiecare comision aplicat are un procentaj (DECIMAL(5,2))

#### ➤ Reparatie

- Id\_reparatie: este identificatorul unic pentru fiecare reparație în parte. Va fi primary key. (INTEGER)
- Numar\_inmatriculare: Asigură relația cu entitatea “Motocicleta”( CHAR(7)).
- Data: va fi NOT NULL deoarece reprezintă data reparației (DATE)
- Cost: va fi NOT NULL deoarece fiecare motocicletă are costuri pentru reparație ( DECIMAL(10,2))

#### ➤ Distribuitor

- Id\_distribuitor: este identificatorul unic pentru fiecare distribuitor în parte. Va fi NOT NULL deoarece fiecare motocicletă are un distribuitor. Va fi primary key. (INT)

- numeDis: va fi NOT NULL, întrucat va conține numele de familie al distribuitorului. Am adăugat o constrângere ce verifică ca numele să fie format doar din litere (VARCHAR (20))
- prenumeDis: va fi NOT NULL, întrucat va conține prenumele distribuitorului. Am adăugat o constrângere ce verifică ca prenumele să fie format doar din litere (VARCHAR (20))
- telefonDis: va fi NOT NULL pentru a salva una din metodele de contact ale distribuitorului și totodată UNIQUE deoarece numerele de telefon inserate sunt românești, deși nu mai adaug prefix acestea vor fi unice pentru fiecare persoană în parte. Pe lângă aceasta am creat o constrângere care să verifice ca numărul de telefon să conțină doar cifre și încă una ce verifică că numărul de telefon conține 10 cifre acesta fiind numărul de cifre pe care le au numerele de telefon românești (VARCHAR (20))
- emailDis: va fi NOT NULL reprezentând metoda de contact de rezervă a clientului și totodată UNIQUE deoarece email-ul este diferit de la individ la individ. Am mai creat o constrângere ce verifică folosind regex formatul mailului pentru a fi unul valid. (VARCHAR (200))
- numar\_inmatriculare: Asigura relația cu entitatea “Motocicleta” ( CHAR(7)).

#### ➤ Imagine

- Id\_img : este identificatorul unic pentru fiecare imagine a produsului în parte. Va fi primary key. ( INTEGER)
- Nume\_img: va fi NOT NULL, întrucât va conține numele imaginii. Am adăugat o constrângere ce verifică ca numele să fie format doar din litere.
- Tip\_img: NOT NULL, format „jpg”, „png”. ( VARCHAR(40))
- Numar\_inmatriculare: Asigură relația cu entitatea “Motocicleta”( CHAR(7)).
- Rezolutie: va fi NOT NULL întrucat reprezintă rezoluția imaginii in pixeli (INTEGER).

#### ➤ Video

- Id\_video: este identificatorul unic pentru fiecare imagine a produsului în parte. Va fi primary key. (INTEGER)
- numeVideo: va fi NOT NULL, întrucât va conține numele imaginii. Am adăugat o constrângere ce verifică ca numele să fie format doar din litere. (VARCHAR(40))
- numar\_inmatriculare: Asigură relația cu entitatea “Motocicleta”( CHAR(7)).
- formatVideo: va fi NOT NULL, întrucât reține formatul videoclipului, care poate fi doar „mp4”, „m4p”, „m4v”. ( INTEGER).



- Video: NOT NULL, conține URL-ul videoclipului. ( VARCHAR(40))

➤ Descriere

- Id\_descriere: este identificatorul unic pentru fiecare descriere a produsului în parte. Va fi primary key.( INTEGER)
- Format\_descriere; va fi NOT NULL, întrucât reține formatul descrierii, care poate fi doar „notepad”, „word”.( VARCHAR(5))
- Descriere: va fi NOT NULL, conține descrierea motocicletei. ( VARCHAR(255))
- Numar\_inmatriculare: Asigură relația cu entitatea “Motocicleta”( CHAR(7)).

➤ Recenzie

- Id\_rececnzie: este identificatorul unic pentru fiecare recenzie a produsului în parte. Va fi primary key.( INTEGER)
- Id\_client: face legatura cu entitatea “Client”. ( INTEGER)
- Recenzie: NOT NULL, conține recenzia.( VARCHAR(200))
- Numar\_inmatriculare: Asigură relația cu entitatea “Motocicleta”( CHAR(7)).

➤ Motociclete\_distribuite

- Numar\_inmatriculare: asigură relația cu entitatea “Motocicleta”( CHAR(7))
- Id\_distribuitoar: asigură relația cu entitatea “Distribuitoar”( INT)
- observatiiMD: dă detalii despre motorul primit de la distribuitor (VARCHAR(250))

➤ Motociclete\_inchiriate

- Numar\_inmatriculare: asigură relația cu entitatea “Motocicleta”( CHAR(7))
- Id\_client: asigură relația cu entitatea “Client”(INTEGER)
- observatiiMI: dă detalii despre motorul primit de client (VARCHAR(200))

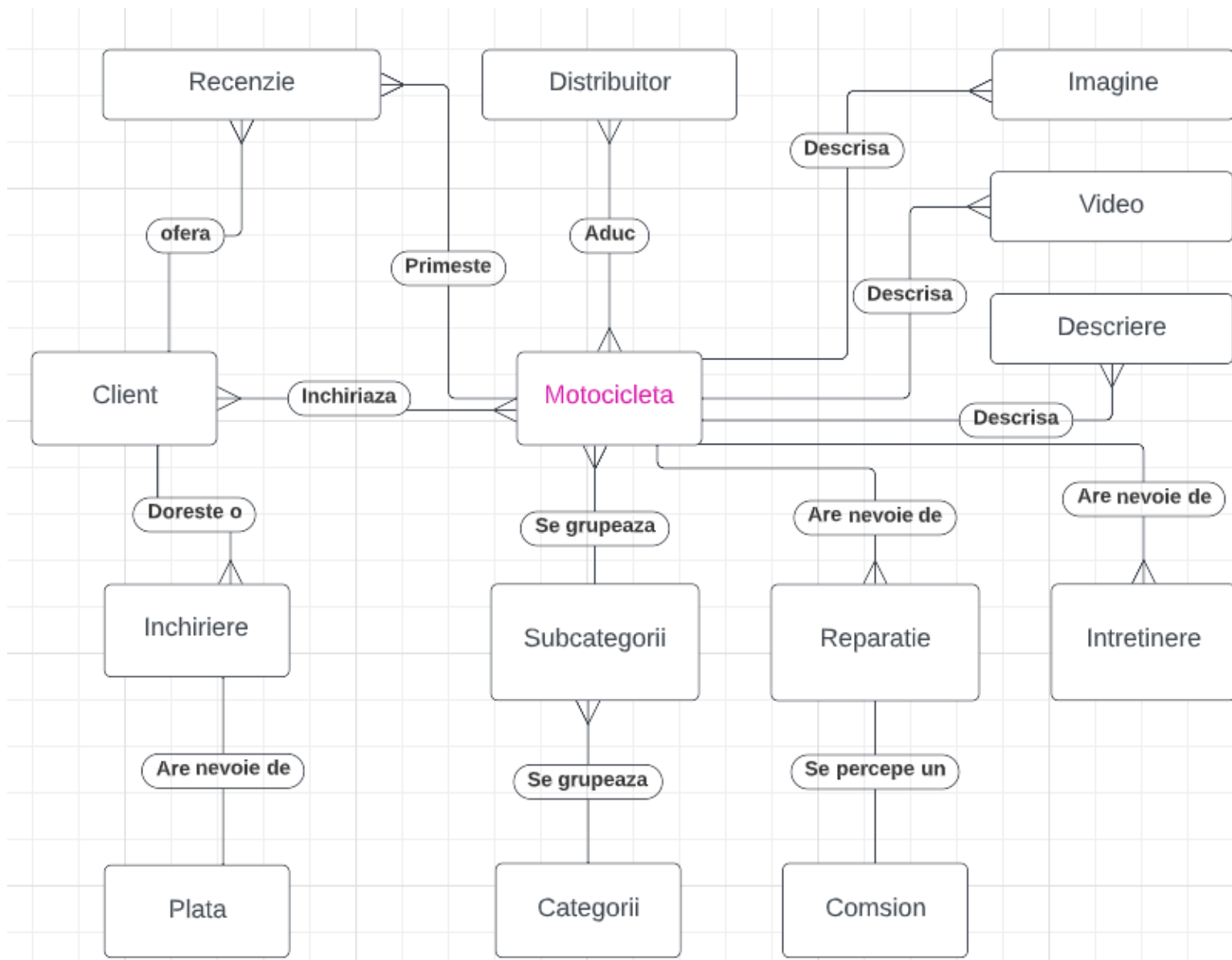
➤ Subcategorii

- Id\_Subcategorii: Identificator unic pentru subcategorii .Va fi primary key (INTEGER).
- numeSubcategorii: NOT NULL. ( VARCHAR(50))
- Id\_categorii: asigură relația cu entitatea “Motocicleta”. ( INTEGER)

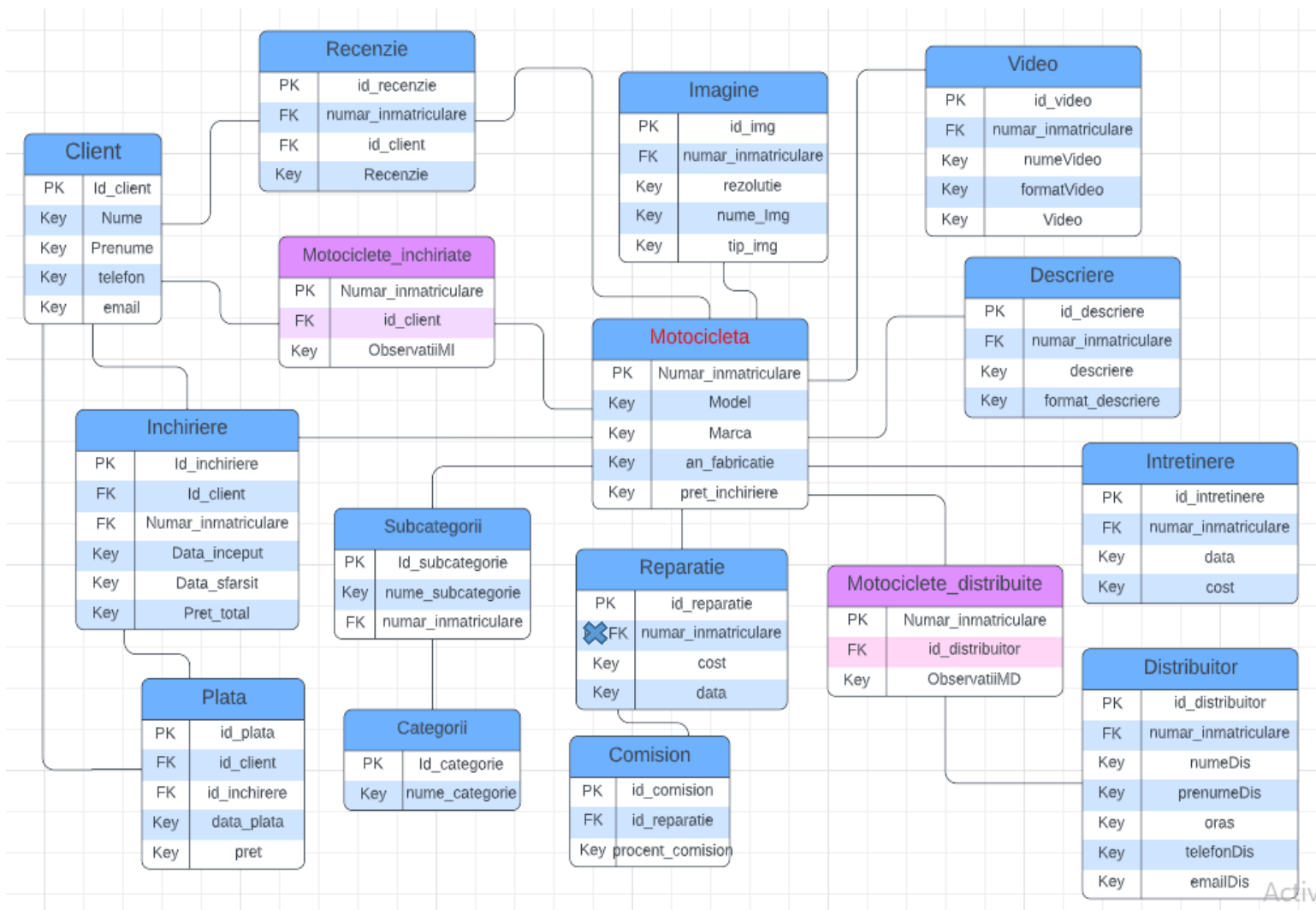
➤ Categorie

- Id\_categorie: Identificator unic pt categorie. Va fi primary key. ( INTEGER)
- numeCategorie: NOT NULL (VARCHAR(50)).

6. Realizarea diagramei entitate-relație corespunzătoare descrierii de la punctele 3-5.



7. Realizarea diagramei conceptuale corespunzătoare diagramei entitate-relație proiectate la punctul 6. Diagrama conceptuală obținută trebuie să conțină minimum 6 tabele (fără considerarea subentităților), dintre care cel puțin un tabel asociativ.



## 8.Enumerarea schemelor relaționale corespunzătoare diagramei conceptuale .

- ❖ Motocicleta(numar\_inmatriculare#,marca,an\_fabricatie,pret\_inchiriere, Id\_subcategorie)
- ❖ Client(id\_client#,nume,prenume,telefon,email,motociclete\_inchiriate)
- ❖ Motociclete\_inchiriate(numar\_inmatriculare#,id\_client#,observatiiMi)
- ❖ Motociclete\_distribuite(numar\_inmatriculare#,observatiiMD,distributor\_id#)
- ❖ Distributor(distributor\_id,numeDis,prenumeDis,telefonDis,emailDis,oras)
- ❖ Reparatie(id\_reparatie#,numar\_inmatriculare#,cost,data)
- ❖ Recenzie(id\_recenzie#,numar\_inmatriculare,recenzie,id\_client)
- ❖ Inchiriere(id\_inchiriere#,id\_client,numar\_inmatriculare,data\_inceput,data\_sfarsit,pret\_total)
- ❖ Categorii(categorii\_id#,numeCategorie)
- ❖ Subcategorie(Id\_subcategorie#,numeSubcategorie,Id\_categorie)
- ❖ Comision(id\_comision#,id\_reparatie,procent\_comision)
- ❖ Descriere(id\_descriere#,numar\_inmatriculare,descriere,format\_descriere)

- ❖ Plata (id\_plata#,id\_client,id\_inchiriere,data\_plata,pret)
- ❖ Video (id\_video#,numar\_inmatriculare,numeVideo,formatAudio,video)
- ❖ Imagine (id\_img#,numar\_inmatriculare,rezolutie,nume\_img,tip\_img)
- ❖ Întreținere (id\_intretinere,numar\_inmatriculare,data,cost)

## 9. Realizarea normalizării până la forma normală 3(FN1-FN3).

Am urmatoarea schemă relațională:

- ❖ Motocicleta(numar\_inmatriculare#,marca,an\_fabricatie,pret\_inchiriere,numeSubc  
ategorie,Id\_categorie)
- ❖ Reparatie (id\_reparatie#,numar\_inmatriculare#,cost,data)
- ❖ Recenzie (id\_recenzie#,numar\_inmatriculare,recenzie,id\_client)
- ❖ Client (id\_client#,nume,prenume,telefon,email,numar\_inmatriculare)
- ❖ Inchiriere(id\_inchiriere#,id\_client,numar\_inmatriculare,data\_inceput,data\_sfarsit,pr  
et\_total)
- ❖ Categori (categori\_id#,numeCategorie)
- ❖ Comision (id\_comision#,id\_reparatie,procent\_comision)
- ❖ Motociclete\_distribuite(numar\_inmatriculare#,observatiiMD,numeDis,prenumeDis,t  
elefonDis,emailDis,oras)
- ❖ Descriere (id\_descriere#,numar\_inmatriculare,descriere,format\_descriere)
- ❖ Plata (id\_plata#,id\_client,id\_inchiriere,data\_plata,pret)
- ❖ Audio (id\_audio#,numar\_inmatriculare,numeAudio,formatAudio,Audio)
- ❖ Video (id\_video#,numar\_inmatriculare,numeVideo,formatAudio,video)
- ❖ Imagine (id\_img#,numar\_inmatriculare,rezolutie,nume\_img,tip\_img)
- ❖ Întreținere (id\_intretinere,numar\_inmatriculare,data,cost)

li aplic FN1 și obțin:

- ❖ Motocicleta (numar\_inmatriculare#,marca,an\_fabricatie,pret\_inchiriere,  
numeSubcategorie,Id\_categorie)
- ❖ Reparație (id\_reparatie#,numar\_inmatriculare#,cost,data)
- ❖ Recenzie (id\_recenzie#,numar\_inmatriculare,recenzie,id\_client)
- ❖ Client (id\_client#,nume,prenume,telefon,email)
- ❖ Motociclete\_inchiriate(numar\_inmatriculare#,id\_client#,observatiiMi)
- ❖ Închiriere(id\_inchiriere#,id\_client,numar\_inmatriculare,data\_inceput,data\_sfarsit,pr  
et\_total)
- ❖ Categori (categori\_id#,numeCategorie)

- ❖ Comision (id\_comision#,id\_reparatie,procent\_comision)
- ❖ Motociclete\_distribuite(numar\_inmatriculare#,observatiiMD,numeDis,prenumeDis,telefonDis,emailDis,oras)
- ❖ Descriere (id\_descriere#,numar\_inmatriculare,descriere,format\_descriere)
- ❖ Plata (id\_plata#,id\_client,id\_inchiriere,data\_plata,pret)
- ❖ Audio (id\_audio#,numar\_inmatriculare,numeAudio,formatAudio,Audio)
- ❖ Video (id\_video#,numar\_inmatriculare,numeVideo,formatAudio,video)
- ❖ Imagine (id\_img#,numar\_inmatriculare,rezolutie,nume\_img,tip\_img)
- ❖ Întreținere (id\_intretinere,numar\_inmatriculare,data,cost)

~Aceasta nu era în FN1, întrucât în tabelul „Client” am coloana „Motociclete\_inchiriate”,căreia nu îi corespunde un atribut indivizibil .Astfel am creat entitatea „Motociclete\_inchiriate”,ca în schema relațională de mai sus.

Îi aplic FN2 și obțin:

- ❖ Motocicleta (numar\_inmatriculare#,marca,an\_fabricatie,pret\_inchiriere,numeSubcategorie,Id\_categorie)
- ❖ Reparație (id\_reparatie#,numar\_inmatriculare#,cost,data)
- ❖ Recenzie (id\_recenzie#,numar\_inmatriculare,recenzie,id\_client)
- ❖ Client (id\_client#,nume,prenume,telefon,email)
- ❖ Motociclete\_inchiriate(numar\_inmatriculare#,id\_client#,observatiiMi)
- ❖ Închiriere(id\_inchiriere#,id\_client,numar\_inmatriculare,data\_inceput,data\_sfarsit,pret\_total)
- ❖ Categori (categorii\_id#,numeCategorie)
- ❖ Comision (id\_comision#,id\_reparatie,procent\_comision)
- ❖ Motociclete\_distribuite (numar\_inmatriculare#,observatiiMD,distribuitor\_id#)
- ❖ Descriere (id\_descriere#,numar\_inmatriculare,descriere,format\_descriere)
- ❖ Plata (id\_plata#,id\_client,id\_inchiriere,data\_plata,pret)
- ❖ Distribuitor(distribuitor\_id,numeDis,prenumeDis,telefonDis,emailDis,oras)
- ❖ Video (id\_video#,numar\_inmatriculare,numeVideo,formatAudio,video)
- ❖ Imagine (id\_img#,numar\_inmatriculare,rezolutie,nume\_img,tip\_img)
- ❖ Întreținere (id\_intretinere,numar\_inmatriculare,data,cost)

~Aceasta nu era în FN2, întrucât în tabelul de „Motociclete\_distribuite” coloanele „numeDis”, „prenumeDis”, „telefonDis”, „emailDis”, „oras” depind doar de „distribuitor\_id”, adică de o parte din cheia primară. Astfel am creat tabelul „Distribuitor”.

Îi aplic FN3 și obțin:

- ❖ Motocicleta (numar\_inmatriculare#,marca,an\_fabricatie,pret\_inchiriere,Id\_subcategorie)
- ❖ Reparatie (id\_reparatie#,numar\_inmatriculare#,cost,data)
- ❖ Recenzie (id\_recenzie#,numar\_inmatriculare,recenzie,id\_client)
- ❖ Client (id\_client#,nume,prenume,telefon,email,motociclete\_inchiriate)
- ❖ Motociclete\_inchiriate(numar\_inmatriculare#,id\_client#,observatiiMi)
- ❖ Inchiriere(id\_inchiriere#,id\_client,numar\_inmatriculare,data\_inceput,data\_sfarsit,pret\_total)
- ❖ Categori (categori\_id#,numeCategorie)
- ❖ Comision (id\_comision#,id\_reparatie,procent\_comision)
- ❖ Motociclete\_distribuite(numar\_inmatriculare#,observatiiMD,distribuitor\_id#)
- ❖ Descriere (id\_descriere#,numar\_inmatriculare,descriere,format\_descriere)
- ❖ Plata (id\_plata#,id\_client,id\_inchiriere,data\_plata,pret)
- ❖ Distribuitor(distribuitor\_id,numeDis,prenumeDis,telefonDis,emailDis,oras)
- ❖ Video (id\_video#,numar\_inmatriculare,numeVideo,formatAudio,video)
- ❖ Imagine (id\_img#,numar\_inmatriculare,rezolutie,nume\_img,tip\_img)
- ❖ Întreținere (id\_intretinere,numar\_inmatriculare,data,cost)
- ❖ Subcategorie(Id\_subcategorie#,numeSubcategorie,Id\_categorie)

~Aceasta nu era în FN3, întrucât în tabelul de „Motociclete” coloana „Id\_categorie” depinde de „numeSubcategorie”, care depinde de „numar\_inmatriculare”. Am rezolvat acest lucru prin crearea entității „Subcategorie”.

**10.** Crearea unei secvențe ce va fi utilizată în inserarea înregistrărilor în tabele (punctul 11).

```
Create table nume_entitate
( atribut1 tipul_variabilei tipul_cheii,
atribut1 tipul_variabilei tipul_cheii,
.....,
atributn tipul_variabilei tipul_cheii,
//si pt FK
FOREIGN KEY (nume2_entitate) REFERENCES nume3_entitate(PK_pt
nume3_entitate)
;
```

**11.** Crearea tabelelor în SQL și inserarea de date coerente în fiecare dintre acestea (minimum 5 înregistrări în fiecare tabel neasociativ; minimum 10 înregistrări în tabelele asociative).

```
CREATE TABLE Client (
ID_client INTEGER PRIMARY KEY,
nume CHAR(255) NOT NULL,
prenume VARCHAR(255) NOT NULL,
email VARCHAR(255) NOT NULL,
telefon CHAR(10) NOT NULL);

CREATE TABLE Motocicleta (
numar_inmatriculare CHAR(7) PRIMARY KEY,
model VARCHAR(255) NOT NULL,
marca VARCHAR(255) NOT NULL,
an_fabricatie INT NOT NULL,
```

```
pret_inchiriere DECIMAL(10,2) NOT NULL,  
CHECK (an_fabricatie >= 1900 AND an_fabricatie <= 2023) );
```

```
CREATE TABLE Inchiriere (  
ID_inchiriere INTEGER PRIMARY KEY,  
ID_client INTEGER NOT NULL,  
numar_inmatriculare CHAR(7) NOT NULL,  
data_inceput DATE NOT NULL,  
data_sfarsit DATE NOT NULL,  
pret_total DECIMAL(10,2) NOT NULL,  
FOREIGN KEY (ID_client) REFERENCES Client(ID_client),  
FOREIGN KEY (numar_inmatriculare) REFERENCES Mototcicleta (numar_inmatriculare),  
CHECK (data_inceput<=data_sfarsit));
```

```
CREATE TABLE Intretinere (  
ID_intretinere INTEGER PRIMARY KEY,  
numar_inmatriculare CHAR(7) NOT NULL,  
data DATE NOT NULL,  
cost DECIMAL(10,2) NOT NULL,  
FOREIGN KEY (numar_inmatriculare) REFERENCES Masina(numar_inmatriculare));
```

```
CREATE TABLE Reparatie (  
ID_reparatie INTEGER PRIMARY KEY,  
numar_inmatriculare CHAR(7) NOT NULL,  
data DATE NOT NULL,  
cost DECIMAL(10,2) NOT NULL,  
FOREIGN KEY (numar_inmatriculare)  
REFERENCES Motocicleta(numar_inmatriculare) );
```



```
CREATE TABLE Comision (  
ID_comision INTEGER PRIMARY KEY,  
ID_reparatie INTEGER NOT NULL,  
procent_comision DECIMAL(5,2) NOT NULL,  
FOREIGN KEY (ID_reparatie)  
REFERENCES Reparatie(ID_reparatie) );
```

```
CREATE TABLE Plata (  
ID_plata INTEGER PRIMARY KEY,  
ID_client INTEGER NOT NULL,  
ID_inchiriere INTEGER NOT NULL,  
data_plata DATE NOT NULL,  
pret DECIMAL(8,2) NOT NULL,  
FOREIGN KEY (ID_client) REFERENCES Client(ID_client),  
FOREIGN KEY (ID_inchiriere)  
REFERENCES Inchiriere(ID_inchiriere) );
```

```
CREATE TABLE Imagine (  
ID_img INTEGER PRIMARY KEY,  
numar_inmatriculare CHAR(7) NOT NULL,  
rezolutie INTEGER NOT NULL,  
nume_img VARCHAR(40) NOT NULL,  
img VARCHAR(40) NOT NULL,  
FOREIGN KEY (numar_inmatriculare)  
REFERENCES Motocicleta(numar_inmatriculare));
```

```
CREATE TABLE Video (  
ID_video INTEGER PRIMARY KEY,
```

```
numar_inmatriculare CHAR(7) NOT NULL,  
formatVideo VARCHAR(7) NOT NULL,  
nume_video VARCHAR(40) NOT NULL,  
video VARCHAR(40) NOT NULL,  
FOREIGN KEY (numar_inmatriculare)  
REFERENCES Motocicleta(numar_inmatriculare));
```

```
CREATE TABLE Descriere (  
ID_descriere INTEGER PRIMARY KEY,  
numar_inmatriculare CHAR(7) NOT NULL,  
descriere VARCHAR(255) NOT NULL,  
format_descriere VARCHAR(5) NOT NULL,  
FOREIGN KEY (numar_inmatriculare)  
REFERENCES Motocicleta(numar_inmatriculare));
```

```
CREATE TABLE Motociclete_inchiriate (  
numar_inmatriculare CHAR(7) NOT NULL,  
ID_client INTEGER NOT NULL,  
observatiiMI VARCHAR(200),  
FOREIGN KEY (numar_inmatriculare)  
REFERENCES Motocicleta(numar_inmatriculare),  
FOREIGN KEY (ID_client)  
REFERENCES Client(ID_client));
```

```
CREATE TABLE Distribuator(  
ID_distribuator INT NOT NULL,  
Numedis VARCHAR (20) NOT NULL,  
Prenumedis VARCHAR (20) NOT NULL,
```

EmailDis VARCHAR (200) NOT NULL,  
TelefonDis VARCHAR (20) NOT NULL,  
numar\_inmatriculare CHAR(7) NOT NULL,  
PRIMARY KEY (ID\_distribuitoar));

CREATE TABLE **Motociclete\_distribuite**(  
numar\_inmatriculare CHAR(7) NOT NULL,  
ID\_distribuitoar INT NOT NULL,  
observatiiMD VARCHAR(250),  
FOREIGN KEY (numar\_inmatriculare)  
REFERENCES Motocicleta(numar\_inmatriculare),  
FOREIGN KEY (ID\_distribuitoar)  
REFERENCES Distribuitoar(ID\_distribuitoar)  
);

CREATE TABLE **Recenzie**(  
ID\_recenzie INTEGER PRIMARY KEY,  
numar\_inmatriculare CHAR(7) NOT NULL,  
ID\_client INTEGER NOT NULL,  
recenzie VARCHAR(200),  
FOREIGN KEY (ID\_client)  
REFERENCES Client(ID\_client),  
FOREIGN KEY (numar\_inmatriculare)  
REFERENCES Motocicleta(numar\_inmatriculare));

CREATE TABLE **Subcategorii**(  
ID\_subcategorii INTEGER PRIMARY KEY,  
numeSubcategorii VARCHAR(50) NOT NULL ,  
ID\_categorii INTEGER,  
FOREIGN KEY (ID\_categorii)

```
REFERENCES Categorie(ID_categorie));
```

```
CREATE TABLE Categorie(  
ID_categorie INTEGER PRIMARY KEY,  
numeCategorie VARCHAR(50) NOT NULL );
```

--Constrângeri pe tabele (anumite constrangeri le-am pus la crearea tabelului)

```
ALTER TABLE Motocicleta  
    ADD CONSTRAINT pret_inchiriere  
    CHECK (pret_inchiriere>0);  
ALTER TABLE Motocicleta  
    ADD CONSTRAINT numar_inmatriculare  
    CHECK (length(numar_inmatriculare)>=6);
```

```
ALTER TABLE Distribuitor  
    ADD CONSTRAINT numeDis  
    CHECK (REGEXP_LIKE(numeDis,'[a-zA-Z]'));  
ALTER TABLE Distribuitor  
    ADD CONSTRAINT prenumeDis  
    CHECK (REGEXP_LIKE(prenumeDis,'[a-zA-Z]'));  
ALTER TABLE Distribuitor  
    ADD CONSTRAINT telefonDis  
    CHECK (REGEXP_LIKE(telefonDis,'[0-9]'));
```

```
ALTER TABLE Clientt  
    ADD CONSTRAINT nume  
    CHECK (REGEXP_LIKE(nume,'[a-zA-Z]'));  
ALTER TABLE Clientt
```

```
ADD CONSTRAINT prenume
CHECK (REGEXP_LIKE(prenume,'[a-zA-Z]'));
ALTER TABLE Clientt
ADD CONSTRAINT telefon
CHECK (REGEXP_LIKE(telefon,'[0-9]'));
```

### --Inserarea de randuri.

```
INSERT INTO Motocicleta (numar_inmatriculare, model, marca, an_fabricatie,
pret_inchiriere)
```

```
VALUES ('B185WHM', 'Golf', 'Volkswagen', 2018, 150);
```

```
INSERT INTO Motocicleta (numar_inmatriculare, model, marca, an_fabricatie,
pret_inchiriere)
```

```
VALUES ('B123FUG', 'Passat', 'Volkswagen', 2013, 120);
```

```
INSERT INTO Motocicleta (numar_inmatriculare, model, marca, an_fabricatie,
pret_inchiriere)
```

```
VALUES ('B07WHM', '1310', 'Dacia', 1997, 60);
```

```
INSERT INTO Motocicleta (numar_inmatriculare, model, marca, an_fabricatie,
pret_inchiriere)
```

```
VALUES ('B01TOP', 'X5', 'BMW', 2010, 175);
```

```
INSERT INTO Motocicleta (numar_inmatriculare, model, marca, an_fabricatie,
pret_inchiriere)
```

```
VALUES ('B106VHJ', 'Scala', 'Skoda', 2022, 150);
```

```
INSERT INTO Motocicleta (numar_inmatriculare, model, marca, an_fabricatie,
pret_inchiriere)
```

```
VALUES ('B06WHM', 'Supernova', 'Dacia', 2001, 90);
```

The screenshot shows a SQL query builder interface. On the left is a tree view of a database schema with tables like CATEGORIE, CLIENT, COMISION, DESCRIERE, DISTRIBUTOR, IMAGINE, INCHIRIERE, INTRETINERE, MOTOCICLETA, MOTOCICLETE\_DISTRIBUT, MOTOCICLETE\_INCHIRIAT, PLATA, RECENZIE, REPARATIE, SUBCATEGORIE, VIDEO, Views, Indexes, Packages, Procedures, Functions, and Operators. The main area shows a query: `select * from Motocicleta;`. Below the query, the 'Query Result' tab displays a table with 6 rows and 5 columns: NUMAR\_INMATRICULARE, MODEL, MARCA, AN\_FABRICATIE, and PRET\_INCHIRIERE.

	NUMAR_INMATRICULARE	MODEL	MARCA	AN_FABRICATIE	PRET_INCHIRIERE
1	B185WHM	Golf	Volkswagen	2018	150
2	B123FUG	Passat	Volkswagen	2013	120
3	B07WHM	1310	Dacia	1997	60
4	B01TOP	X5	BMW	2010	175
5	B106VHJ	Scala	Skoda	2022	150
6	B06WHM	Supernova	Dacia	2001	90

INSERT INTO Inchiriere (ID\_inchiriere, ID\_client, numar\_inmatriculare, data\_inceput, data\_sfarsit, pret\_total)

VALUES (1, 1, 'B185WHM', TO\_DATE('2022-02-03', 'YYYY-MM-DD'), TO\_DATE('2022-02-07', 'YYYY-MM-DD'), 840);

INSERT INTO Inchiriere (ID\_inchiriere, ID\_client, numar\_inmatriculare, data\_inceput, data\_sfarsit, pret\_total)

VALUES (2, 4, 'B06WHM', TO\_DATE('2022-05-09', 'YYYY-MM-DD'), TO\_DATE('2022-06-24', 'YYYY-MM-DD'), 1496.25);

INSERT INTO Inchiriere (ID\_inchiriere, ID\_client, numar\_inmatriculare, data\_inceput, data\_sfarsit, pret\_total)

VALUES (3, 2, 'B123FUG', TO\_DATE('2022-10-01', 'YYYY-MM-DD'), TO\_DATE('2022-10-07', 'YYYY-MM-DD'), 498.54);

INSERT INTO Inchiriere (ID\_inchiriere, ID\_client, numar\_inmatriculare, data\_inceput, data\_sfarsit, pret\_total)

VALUES (4, 1, 'B185WHM', TO\_DATE('2022-11-02', 'YYYY-MM-DD'), TO\_DATE('2022-12-01', 'YYYY-MM-DD'), 974.20);

INSERT INTO Inchiriere (ID\_inchiriere, ID\_client, numar\_inmatriculare, data\_inceput, data\_sfarsit, pret\_total)

VALUES (5, 2, 'B185WHM', TO\_DATE('2022-09-15', 'YYYY-MM-DD'), TO\_DATE('2022-09-30', 'YYYY-MM-DD'), 758.50);

INSERT INTO Inchiriere (ID\_inchiriere, ID\_client, numar\_inmatriculare, data\_inceput, data\_sfarsit, pret\_total)

VALUES (6, 1, 'B185WHM', TO\_DATE('2022-07-09', 'YYYY-MM-DD'), TO\_DATE('2022-07-14', 'YYYY-MM-DD'), 456.12);

INSERT INTO Inchiriere (ID\_inchiriere, ID\_client, numar\_inmatriculare, data\_inceput, data\_sfarsit, pret\_total)

VALUES (7, 1, 'B01TOP', TO\_DATE('2022-02-07', 'YYYY-MM-DD'), TO\_DATE('2022-02-15', 'YYYY-MM-DD'), 550);

INSERT INTO Inchiriere (ID\_inchiriere, ID\_client, numar\_inmatriculare, data\_inceput, data\_sfarsit, pret\_total)

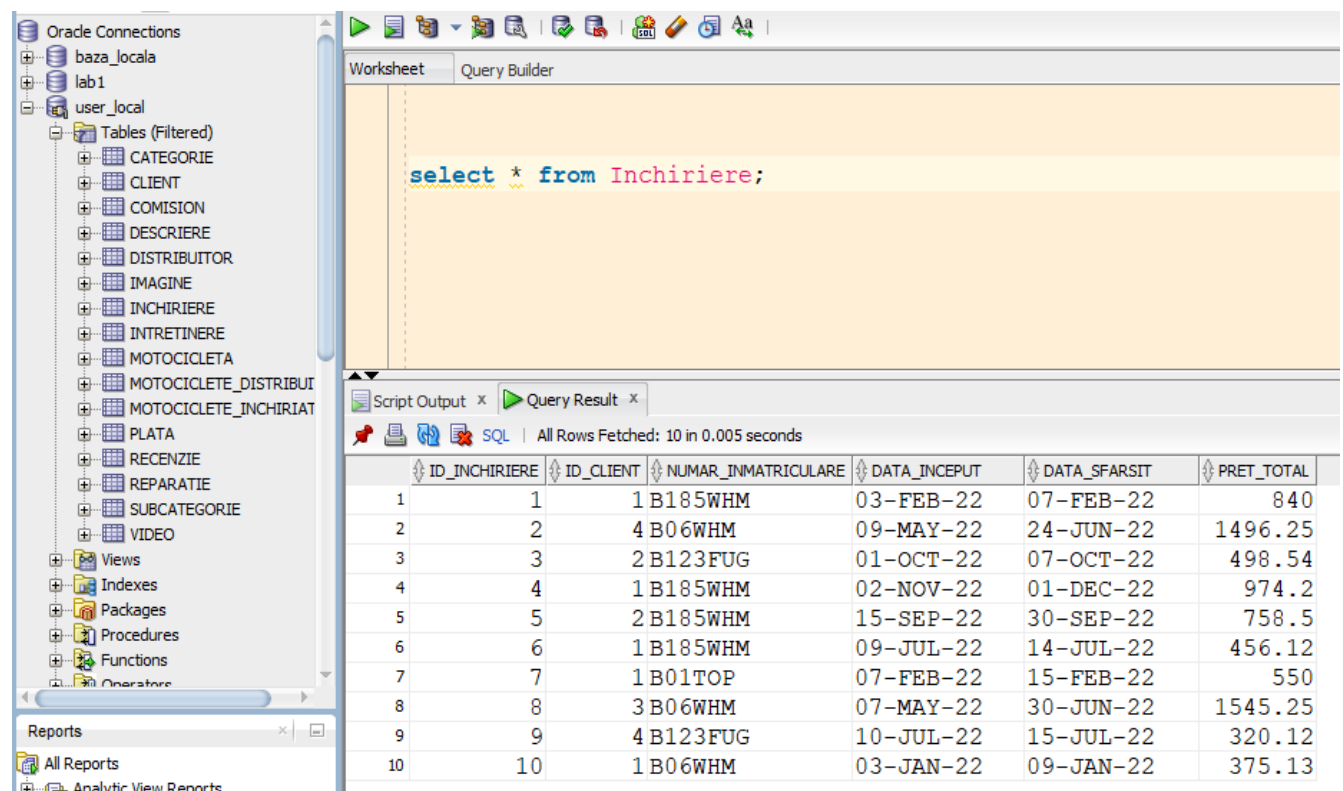
VALUES (8, 3, 'B06WHM', TO\_DATE('2022-05-07', 'YYYY-MM-DD'), TO\_DATE('2022-06-30', 'YYYY-MM-DD'), 1545.25);

INSERT INTO Inchiriere (ID\_inchiriere, ID\_client, numar\_inmatriculare, data\_inceput, data\_sfarsit, pret\_total)

VALUES (9, 4, 'B123FUG', TO\_DATE('2022-07-10', 'YYYY-MM-DD'), TO\_DATE('2022-07-15', 'YYYY-MM-DD'), 320.12);

INSERT INTO Inchiriere (ID\_inchiriere, ID\_client, numar\_inmatriculare, data\_inceput, data\_sfarsit, pret\_total)

VALUES (10, 1, 'B06WHM', TO\_DATE('2022-01-03', 'YYYY-MM-DD'), TO\_DATE('2022-01-09', 'YYYY-MM-DD'), 375.13);



The screenshot shows the Oracle SQL Developer interface. On the left, the 'Object Explorer' pane displays the database structure for 'baza\_locala', including tables like CATEGORIE, CLIENT, COMISION, DESCRIE, DISTRIBUTOR, IMAGINE, INCHIRIERE, INTRETINERE, MOTOCICLETA, MOTOCICLETE\_DISTRIBUT, MOTOCICLETE\_INCHIRIAT, PLATA, RECENZIE, REPARATIE, SUBCATEGORIE, and VIDEO. The 'Query Builder' pane in the center contains the SQL query: `select * from Inchiriere;`. The 'Query Result' pane at the bottom displays the results of the query, showing 10 rows of data. The columns are ID\_INCHIRIERE, ID\_CLIENT, NUMAR\_INMATRICULARE, DATA\_INCEPUT, DATA\_SFARSIT, and PRET\_TOTAL.

ID_INCHIRIERE	ID_CLIENT	NUMAR_INMATRICULARE	DATA_INCEPUT	DATA_SFARSIT	PRET_TOTAL
1	1	1 B185WHM	03-FEB-22	07-FEB-22	840
2	2	4 B06WHM	09-MAY-22	24-JUN-22	1496.25
3	3	2 B123FUG	01-OCT-22	07-OCT-22	498.54
4	4	1 B185WHM	02-NOV-22	01-DEC-22	974.2
5	5	2 B185WHM	15-SEP-22	30-SEP-22	758.5
6	6	1 B185WHM	09-JUL-22	14-JUL-22	456.12
7	7	1 B01TOP	07-FEB-22	15-FEB-22	550
8	8	3 B06WHM	07-MAY-22	30-JUN-22	1545.25
9	9	4 B123FUG	10-JUL-22	15-JUL-22	320.12
10	10	1 B06WHM	03-JAN-22	09-JAN-22	375.13

```
INSERT INTO Intretinere (ID_intretinere, numar_inmatriculare, data, cost)
VALUES (1, 'B185WHM', TO_DATE('2022-12-20', 'YYYY-MM-DD'), 100);
INSERT INTO Intretinere (ID_intretinere, numar_inmatriculare, data, cost)
VALUES (2, 'B185WHM', TO_DATE('2022-09-12', 'YYYY-MM-DD'), 250);
INSERT INTO Intretinere (ID_intretinere, numar_inmatriculare, data, cost)
VALUES (3, 'B06WHM', TO_DATE('2022-10-01', 'YYYY-MM-DD'), 150);
INSERT INTO Intretinere (ID_intretinere, numar_inmatriculare, data, cost)
VALUES (4, 'B123FUG', TO_DATE('2022-11-25', 'YYYY-MM-DD'), 25);
INSERT INTO Intretinere (ID_intretinere, numar_inmatriculare, data, cost)
VALUES (5, 'B185WHM', TO_DATE('2022-05-20', 'YYYY-MM-DD'), 120);
INSERT INTO Intretinere (ID_intretinere, numar_inmatriculare, data, cost)
VALUES (6, 'B07WHM', TO_DATE('2022-12-20', 'YYYY-MM-DD'), 100);
INSERT INTO Intretinere (ID_intretinere, numar_inmatriculare, data, cost)
VALUES (7, 'B01TOP', TO_DATE('2022-12-20', 'YYYY-MM-DD'), 190);
INSERT INTO Intretinere (ID_intretinere, numar_inmatriculare, data, cost)
VALUES (8, 'B185WHM', TO_DATE('2022-05-10', 'YYYY-MM-DD'), 300);
INSERT INTO Intretinere (ID_intretinere, numar_inmatriculare, data, cost)
VALUES (9, 'B123FUG', TO_DATE('2022-01-05', 'YYYY-MM-DD'), 70);
INSERT INTO Intretinere (ID_intretinere, numar_inmatriculare, data, cost)
VALUES (10, 'B06WHM', TO_DATE('2022-01-15', 'YYYY-MM-DD'), 30);
```



The screenshot shows the Oracle SQL Developer interface. On the left, the 'Oracle Connections' tree shows a connection to 'baza\_locala'. The 'Tables (Filtered)' list includes: CATEGORIE, CLIENT, COMISION, DESCRIERE, DISTRIBUTOR, IMAGINE, INCHIRIERE, INTRETINERE, MOTOCICLETA, MOTOCICLETE\_DISTRIBUI, MOTOCICLETE\_INCHIRIAT, PLATA, RECENZIE, REPARATIE, SUBCATEGORIE, and VIDEO. The 'Query Builder' window shows the query: `select * from Intretinere;`. The 'Query Result' window shows the following data:

ID_INTRETINERE	NUMAR_INMATRICULARE	DATA	COST
1	1 B185WHM	20-DEC-22	100
2	2 B185WHM	12-SEP-22	250
3	3 B06WHM	01-OCT-22	150
4	4 B123FUG	25-NOV-22	25
5	5 B185WHM	20-MAY-22	120
6	6 B07WHM	20-DEC-22	100
7	7 B01TOP	20-DEC-22	190
8	8 B185WHM	10-MAY-22	300
9	9 B123FUG	05-JAN-22	70
10	10 B06WHM	15-JAN-22	30

```

INSERT INTO Reparatie (ID_reparatie, numar_inmatriculare, data, cost)
VALUES (1, 'B123FUG', TO_DATE('2022-01-15', 'YYYY-MM-DD'), 200);

INSERT INTO Reparatie (ID_reparatie, numar_inmatriculare, data, cost)
VALUES (2, 'B185WHM', TO_DATE('2022-01-30', 'YYYY-MM-DD'), 500);

INSERT INTO Reparatie (ID_reparatie, numar_inmatriculare, data, cost)
VALUES (3, 'B06WHM', TO_DATE('2022-12-04', 'YYYY-MM-DD'), 350);

INSERT INTO Reparatie (ID_reparatie, numar_inmatriculare, data, cost)
VALUES (4, 'B185WHM', TO_DATE('2022-11-24', 'YYYY-MM-DD'), 150);

INSERT INTO Reparatie (ID_reparatie, numar_inmatriculare, data, cost)
VALUES (5, 'B06WHM', TO_DATE('2022-10-01', 'YYYY-MM-DD'), 300);

```

The screenshot shows the Oracle SQL Developer interface. On the left, the 'Orade Connections' tree is expanded to 'user\_local', showing a list of tables including 'REPARATIE'. The main window is the 'Query Builder' tab, displaying the SQL query: `select * from Reparatie;`. Below the query, the 'Query Result' tab shows the results of the query. The results are displayed in a table with 5 rows and 4 columns: ID\_REPARATIE, NUMAR\_INMATRICULARE, DATA, and COST.

	ID_REPARATIE	NUMAR_INMATRICULARE	DATA	COST
1	1	B123FUG	15-JAN-22	200
2	2	B185WHM	30-JAN-22	500
3	3	B06WHM	04-DEC-22	350
4	4	B185WHM	24-NOV-22	150
5	5	B06WHM	01-OCT-22	300

```

INSERT INTO Clientt(ID_client,nume,prenume,email,telefon)
VALUES (1,'Ion','Adina','adinaion53@gmail.com','0799790528');

INSERT INTO Clientt (ID_client, nume, prenume, email, telefon)
VALUES (2, 'Popescu', 'Ion', 'ion.popescu@gmail.com', '0712345678');

INSERT INTO Clientt (ID_client, nume, prenume, email, telefon)
VALUES (3, 'Mihai', 'Ioana', 'ioana.mihai@yahoo.com', '0799887766');

INSERT INTO Clientt (ID_client, nume, prenume, email, telefon)
VALUES (4, 'Grigore', 'Alexandru', 'alex.grigore@gmail.com', '0712345999');

INSERT INTO Clientt (ID_client, nume, prenume, email, telefon)
VALUES (5, 'Argesanu', 'Rodica', 'argesanu.r@yahoo.com', '0778546123');

INSERT INTO Clientt (ID_client, nume, prenume, email, telefon)
VALUES (6, 'Oprea', 'Tudor', 'tudor.oprea@hotmail.com', '0710986338');

INSERT INTO Clientt (ID_client, nume, prenume, email, telefon)
VALUES (7, 'Ion', 'Andy', 'ion.andy@yahoo.com', '079931766');

INSERT INTO Clientt (ID_client, nume, prenume, email, telefon)

```

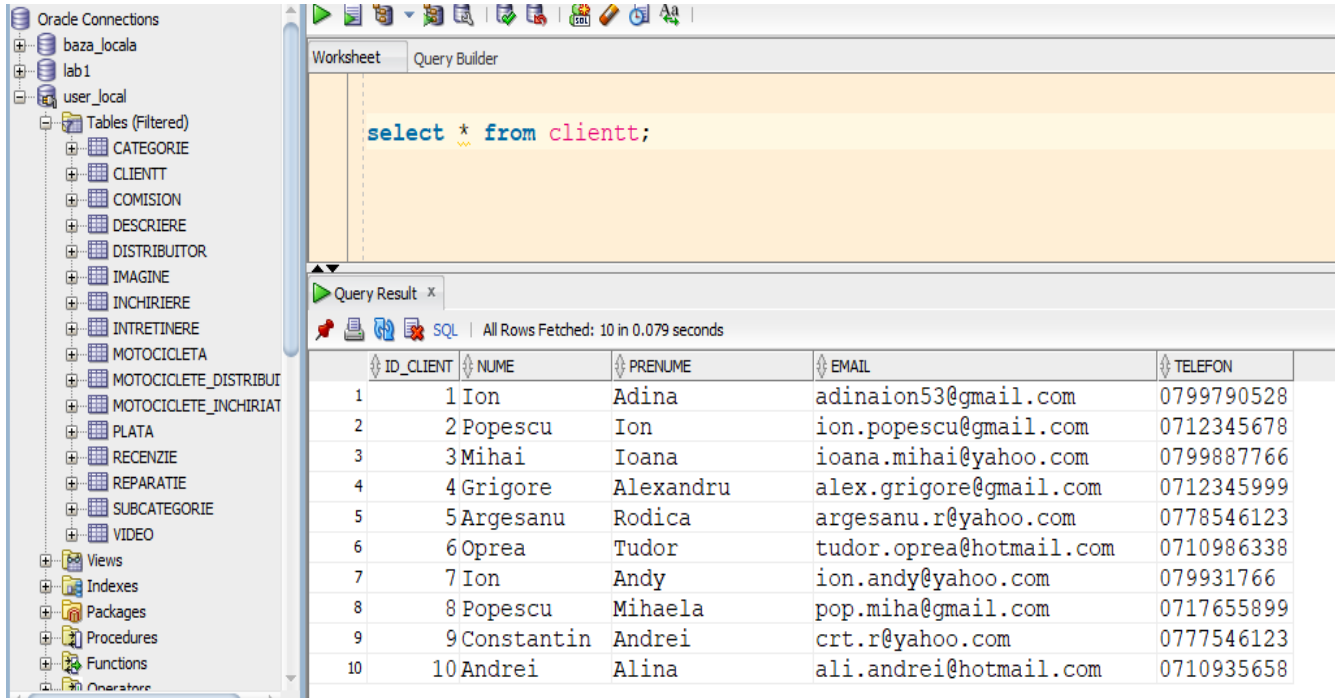
```
VALUES (8, 'Popescu', 'Mihaela', 'pop.miha@gmail.com', '071755899');
```

```
INSERT INTO Clientt (ID_client, nume, prenume, email, telefon)
```

```
VALUES (9, 'Constantin', 'Andrei', 'crt.r@yahoo.com', '0777546123');
```

```
INSERT INTO Clientt (ID_client, nume, prenume, email, telefon)
```

```
VALUES (10, 'Andrei', 'Alina', 'ali.andrei@hotmail.com', '0710935658');
```



The screenshot shows the Oracle SQL Developer interface. On the left, the 'Tables (Filtered)' list includes: CATEGORIE, CLIENTT, COMISION, DESCRIERE, DISTRIBUTOR, IMAGINE, INCHIRIERE, INTRETINERE, MOTOCICLETA, MOTOCICLETA\_DISTRIBUI, MOTOCICLETA\_INCHIRIAT, PLATA, RECENZIE, REPARATIE, SUBCATEGORIE, and VIDEO. The main window displays a query: `select * from clientt;`. Below the query, the 'Query Result' tab shows the data for the 'clientt' table. The result is a table with 10 rows and 5 columns: ID\_CLIENT, NUME, PRENUME, EMAIL, and TELEFON.

ID_CLIENT	NUME	PRENUME	EMAIL	TELEFON
1	1 Ion	Adina	adinaion53@gmail.com	0799790528
2	2 Popescu	Ion	ion.popescu@gmail.com	0712345678
3	3 Mihai	Ioana	ioana.mihai@yahoo.com	0799887766
4	4 Grigore	Alexandru	alex.grigore@gmail.com	0712345999
5	5 Argesanu	Rodica	argesanu.r@yahoo.com	0778546123
6	6 Oprea	Tudor	tudor.oprea@hotmail.com	0710986338
7	7 Ion	Andy	ion.andy@yahoo.com	079931766
8	8 Popescu	Mihaela	pop.miha@gmail.com	0717655899
9	9 Constantin	Andrei	crt.r@yahoo.com	0777546123
10	10 Andrei	Alina	ali.andrei@hotmail.com	0710935658

```
INSERT INTO Plata (ID_plata , ID_client , ID_inchiriere , data_plata , pret )
```

```
VALUES (1, 1, 1, TO_DATE('2022-02-07', 'YYYY-MM-DD'), 840);
```

```
INSERT INTO Plata (ID_plata , ID_client , ID_inchiriere , data_plata , pret )
```

```
VALUES (2, 4, 2, TO_DATE('2022-06-24', 'YYYY-MM-DD'), 1496.25);
```

```
INSERT INTO Plata (ID_plata , ID_client , ID_inchiriere , data_plata , pret )
```

```
VALUES (3, 2, 3, TO_DATE('2022-10-07', 'YYYY-MM-DD'), 498.54);
```

```
INSERT INTO Plata (ID_plata , ID_client , ID_inchiriere , data_plata , pret )
```

```
VALUES (4, 1, 2, TO_DATE('2022-12-01', 'YYYY-MM-DD'), 974.20);
```

```
INSERT INTO Plata (ID_plata , ID_client , ID_inchiriere , data_plata , pret )
```

```
VALUES (5, 2, 2, TO_DATE('2022-09-30', 'YYYY-MM-DD'), 758.5);
```

```
INSERT INTO Plata (ID_plata , ID_client , ID_inchiriere , data_plata , pret )
```

```
VALUES (6, 1, 1, TO_DATE('2022-07-14', 'YYYY-MM-DD'), 456.12);
```

```
INSERT INTO Plata (ID_plata , ID_client , ID_inchiriere , data_plata , pret )
```

```
VALUES (7, 1, 3, TO_DATE('2022-02-15', 'YYYY-MM-DD'), 550);
```

```
INSERT INTO Plata (ID_plata , ID_client , ID_inchiriere , data_plata , pret )
```

```
VALUES (8, 3, 2, TO_DATE('2022-06-30', 'YYYY-MM-DD'), 1545.25);
```

```
INSERT INTO Plata (ID_plata , ID_client , ID_inchiriere , data_plata , pret )
```

```
VALUES (9, 4, 2, TO_DATE('2022-07-15', 'YYYY-MM-DD'), 320.12);
```

```
INSERT INTO Plata (ID_plata , ID_client , ID_inchiriere , data_plata , pret )
```

```
VALUES (10, 1, 3, TO_DATE('2022-01-09', 'YYYY-MM-DD'), 375.13);
```

The screenshot shows the Oracle SQL Developer interface. On the left, the 'Oracle Connections' tree is expanded to 'user\_local' > 'Tables (Filtered)', showing a list of tables including PLATA. The main window is in 'Query Builder' mode, displaying the SQL query: `select * from Plata;`. Below the query, the 'Query Result' tab shows the data fetched from the table. The result is a table with 10 rows and 5 columns: ID\_PLATA, ID\_CLIENT, ID\_INCHIRIERE, DATA\_PLATA, and PRET.

ID_PLATA	ID_CLIENT	ID_INCHIRIERE	DATA_PLATA	PRET
1	1	1	1 07-FEB-22	840
2	2	4	2 24-JUN-22	1496.25
3	3	2	3 07-OCT-22	498.54
4	4	1	4 01-DEC-22	974.2
5	5	2	5 30-SEP-22	758.5
6	6	1	6 14-JUL-22	456.12
7	7	1	7 15-FEB-22	550
8	8	3	8 30-JUN-22	1545.25
9	9	4	9 15-JUL-22	320.12
10	10	1	10 09-JAN-22	375.13

```
INSERT INTO Comision (ID_comision, ID_reparatie, procent_comision)
```

```
VALUES (1, 1, 10);
```

```
INSERT INTO Comision (ID_comision, ID_reparatie, procent_comision)
```

```
VALUES (2, 4 , 15);
```

```
INSERT INTO Comision (ID_comision, ID_reparatie, procent_comision)
```

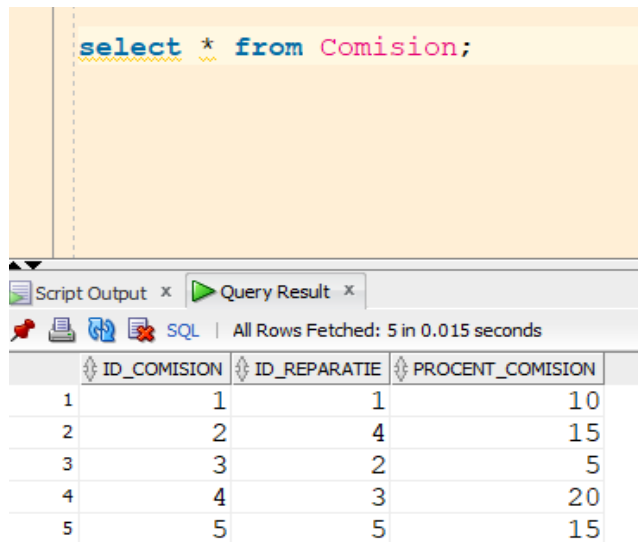
```
VALUES (3, 2 , 5);
```

```
INSERT INTO Comision (ID_comision, ID_reparatie, procent_comision)
```

```
VALUES (4, 3 , 20);
```

```
INSERT INTO Comision (ID_comision, ID_reparatie, procent_comision)
```

VALUES (5, 5 , 15);



The screenshot shows a SQL query execution window. The query entered is `select * from Comision;`. The results are displayed in a table with 5 rows. The table has three columns: `ID_COMISION`, `ID_REPARATIE`, and `PROCENT_COMISION`. The data is as follows:

ID_COMISION	ID_REPARATIE	PROCENT_COMISION
1	1	10
2	2	15
3	3	5
4	4	20
5	5	15

INSERT INTO Imagine (Id\_img,numar\_inmatriculare,rezolutie,nume\_img,img)  
VALUES (1,'B185WHM',2000,'img\_B185WHM','https://ur12wl');

INSERT INTO Imagine (Id\_img,numar\_inmatriculare,rezolutie,nume\_img,img)  
VALUES (2,'B06WHM',2001,'img\_B06WHM','https://uasdasrl');

INSERT INTO Imagine (Id\_img,numar\_inmatriculare,rezolutie,nume\_img,img)  
VALUES (3,'B123FUG',2002,'img\_B123FUG','https://uasdasrl');

INSERT INTO Imagine (Id\_img,numar\_inmatriculare,rezolutie,nume\_img,img)  
VALUES (4,'B01TOP',2002,'img\_B01TOP','https://uas66dasrl');

INSERT INTO Imagine (Id\_img,numar\_inmatriculare,rezolutie,nume\_img,img)  
VALUES (5,'B185WHM',2002,'img\_B01TOP','https://uas366dasrl');

INSERT INTO Imagine (Id\_img,numar\_inmatriculare,rezolutie,nume\_img,img)  
VALUES (6,'B185WHM',2000,'img\_B185WHM','https://ur12wl');

INSERT INTO Imagine (Id\_img,numar\_inmatriculare,rezolutie,nume\_img,img)  
VALUES (7,'B01TOP',2002,'img\_B01TOP','https://uas66dasrl');

INSERT INTO Imagine (Id\_img,numar\_inmatriculare,rezolutie,nume\_img,img)  
VALUES (8,'B123FUG',25502,'img\_B123FUG','https://ua234sdasrl');

INSERT INTO Imagine (Id\_img,numar\_inmatriculare,rezolutie,nume\_img,img)  
VALUES (9,'B07WHM',2001,'img\_B06WHM','https://uasdbsebasrl');

The screenshot shows the SQL Developer interface. On the left, the 'Connections' pane lists 'baza\_locala', 'lab1', and 'user\_local'. The 'user\_local' connection is selected, showing a list of tables including 'CATEGORIE', 'CLIENT', 'COMISION', 'DESCRIERE', 'DISTRIBUTOR', 'IMAGINE', 'INCHIRIERE', 'INTRETINERE', 'MOTOCICLETA', 'MOTOCICLETE\_DISTRIBUIT', 'MOTOCICLETE\_INCHIRIAT', 'PLATA', 'RECENZIE', 'REPARATIE', 'SUBCATEGORIE', and 'VIDEO'. The main window displays a SQL worksheet with the query: `select * from Imagine;`. Below the worksheet, the 'Query Result' pane shows the results of the query, with 9 rows fetched in 0.013 seconds. The results are as follows:

ID_IMG	NUMAR_INMATRICULARE	REZOLUTIE	NUME_IMG	IMG
1	B185WHM	2000	img B185WHM	https://url
2	B06WHM	2001	img B06WHM	https://url
3	B123FUG	2002	img B123FUG	https://url
4	B01TOP	2002	img B01TOP	https://uas66dasrl
5	B185WHM	2002	img B01TOP	https://uas366dasrl
6	B185WHM	2000	img B185WHM	https://url2wl
7	B01TOP	2002	img B01TOP	https://uas66dasrl
8	B123FUG	25502	img B123FUG	https://ua234sdasrl
9	B07WHM	2001	img B06WHM	https://uasdbsebasrl

```
INSERT INTO Video (Id_video,numar_inmatriculare,formatVideo,nume_video,video)
VALUES (1,'B185WHM',3,'vid1','https://ureqwfwl');
```

```
INSERT INTO Video (Id_video,numar_inmatriculare,formatVideo,nume_video,video)
VALUES (2,'B06WHM',43,'vid1','https://ur12wdewfrgthtjyl');
```

```
INSERT INTO Video (Id_video,numar_inmatriculare,formatVideo,nume_video,video)
VALUES (3,'B06WHM',63,'vid2','https://ur12wwe32l');
```

```
INSERT INTO Video (Id_video,numar_inmatriculare,formatVideo,nume_video,video)
VALUES (4,'B123FUG',38,'vid3','https://ur12ffffwl');
```

```
INSERT INTO Video (Id_video,numar_inmatriculare,formatVideo,nume_video,video)
VALUES (5,'B123FUG',3,'vid4','https://ur12whyjyl');
```

```
INSERT INTO Video (Id_video,numar_inmatriculare,formatVideo,nume_video,video)
VALUES (6,'B123FUG',93,'vid5','https://ur12wooul');
```

```
INSERT INTO Video (Id_video,numar_inmatriculare,formatVideo,nume_video,video)
```

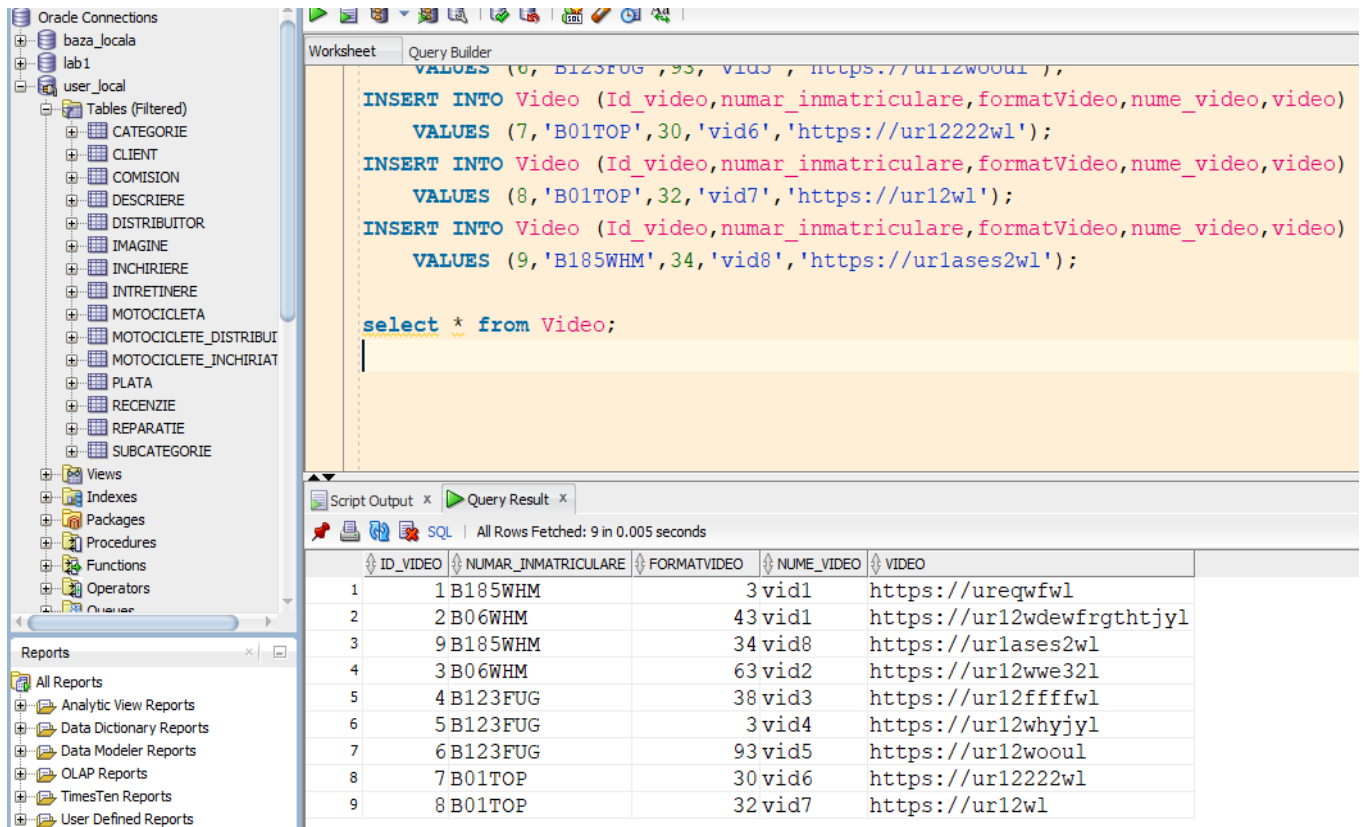
```
VALUES (7,'B01TOP',30,'vid6','https://ur12222wl');
```

```
INSERT INTO Video (Id_video,numar_inmatriculare,formatVideo,nume_video,video)
```

```
VALUES (8,'B01TOP',32,'vid7','https://ur12wl');
```

```
INSERT INTO Video (Id_video,numar_inmatriculare,formatVideo,nume_video,video)
```

```
VALUES (9,'B185WHM',34,'vid8','https://ur1ases2wl');
```



The screenshot shows the Oracle SQL Developer interface. The left pane displays the database schema with tables like CATEGORIE, CLIENT, COMISION, DESCRIERE, DISTRIBUTOR, IMAGINE, INCHIRIERE, INTRETINERE, MOTOCICLETA, MOTOCICLETE\_DISTRIBUT, MOTOCICLETE\_INCHIRIAT, PLATA, RECENZIE, REPARATIE, and SUBCATEGORIE. The main query window contains the following SQL code:

```
VALUES (6, 'B123FUG', 93, 'vid5', 'https://ur12woou1' );
INSERT INTO Video (Id_video,numar_inmatriculare,formatVideo,nume_video,video)
VALUES (7, 'B01TOP', 30, 'vid6', 'https://ur12222wl');
INSERT INTO Video (Id_video,numar_inmatriculare,formatVideo,nume_video,video)
VALUES (8, 'B01TOP', 32, 'vid7', 'https://ur12wl');
INSERT INTO Video (Id_video,numar_inmatriculare,formatVideo,nume_video,video)
VALUES (9, 'B185WHM', 34, 'vid8', 'https://ur1ases2wl');

select * from Video;
```

The Query Result window shows the following data:

ID_VIDEO	NUMAR_INMATRICULARE	FORMATVIDEO	NUME_VIDEO	VIDEO
1	1 B185WHM		3 vid1	https://ureqwfwl
2	2 B06WHM		43 vid1	https://ur12wdewfrgthtjyl
3	9 B185WHM		34 vid8	https://ur1ases2wl
4	3 B06WHM		63 vid2	https://ur12wwe32l
5	4 B123FUG		38 vid3	https://ur12ffffwl
6	5 B123FUG		3 vid4	https://ur12whyjyl
7	6 B123FUG		93 vid5	https://ur12wooul
8	7 B01TOP		30 vid6	https://ur12222wl
9	8 B01TOP		32 vid7	https://ur12wl

```
INSERT INTO Descriere (Id_descriere,numar_inmatriculare,descriere,format_descriere)
```

```
VALUES (1,'B185WHM','Motor 2.1','h');
```

```
INSERT INTO Descriere (Id_descriere,numar_inmatriculare,descriere,format_descriere)
```

```
VALUES (2,'B06WHM','Viteza medie 100km/h','hs');
```

```
INSERT INTO Descriere (Id_descriere,numar_inmatriculare,descriere,format_descriere)
```

```
VALUES (3,'B06WHM','Precizie la viraje','3');
```



```

INSERT INTO Descriere(Id_descriere,numar_inmatriculare,descriere,format_descriere)
VALUES (4,'B123FUG','Suspensii bune','hl');

INSERT INTO Descriere (Id_descriere,numar_inmatriculare,descriere,format_descriere)
VALUES (5,'B123FUG','Faruri omologate','hl');

INSERT INTO Descriere (Id_descriere,numar_inmatriculare,descriere,format_descriere)
VALUES (6,'B123FUG','Miniportbagaj','ht');

INSERT INTO Descriere (Id_descriere,numar_inmatriculare,descriere,format_descriere)
VALUES (7,'B01TOP','Roti de vara noi','h');

INSERT INTO Descriere (Id_descriere,numar_inmatriculare,descriere,format_descriere)
VALUES (8,'B01TOP','Motor 1.5','rrr');

INSERT INTO Descriere (Id_descriere,numar_inmatriculare,descriere,format_descriere)
VALUES (9,'B185WHM','Motor Diesel','ht');

```

Oracle SQL Developer : C:\Users\Adina\user\_local2.sql

File Edit View Navigate Run Source Team Tools Window Help

Connections

- Oracle Connections
  - baza\_locala
  - lab1
  - user\_local
    - Tables (Filtered)
      - CATEGORIE
      - CLIENT
      - COMISION
      - DESCRIERE
      - DISTRIBUTOR
      - IMAGINE
      - INCHIRIERE
      - INTRETINERE
      - MOTOCICLETA
      - MOTOCICLETE\_DISTRIBUI
      - MOTOCICLETE\_INCHIRIAT
      - PLATA
      - RECENZIE
      - REPARATIE
      - SUBCATEGORIE
    - Views
    - Indexes
    - Packages
    - Procedures
    - Functions
    - Operators
    - Oueues

Reports

- All Reports
- Analytic View Reports
- Data Dictionary Reports
- Data Modeler Reports
- OLAP Reports
- TimesTen Reports
- User Defined Reports

user\_local2.sql

SQL Worksheet History

Worksheet Query Builder

```

INSERT INTO Descriere (Id_descriere,numar_inmatriculare,descriere,
VALUES (7,'B01TOP','Roti de vara noi','h');
INSERT INTO Descriere (Id_descriere,numar_inmatriculare,descriere,
VALUES (8,'B01TOP','Motor 1.5','rrr');
INSERT INTO Descriere (Id_descriere,numar_inmatriculare,descriere,
VALUES (9,'B185WHM','Motor Diesel','ht');

select * from descriere;

```

Script Output x Query Result x

SQL | All Rows Fetched: 9 in 0.004 seconds

ID_DESCRIERE	NUMAR_INMATRICULARE	DESCRIERE	FORMAT_DESCRIERE
1	2 B06WHM	Viteza medie 100km/h	https
2	5 B123FUG	Faruri omologate	httl
3	6 B123FUG	Miniportbagaj	ht
4	7 B01TOP	Roti de vara noi	h
5	8 B01TOP	Motor 1.5	rrr
6	9 B185WHM	Motor Diesel	ht
7	1 B185WHM	Motor 2.1	ht
8	3 B06WHM	Precizie la viraje	3
9	4 B123FUG	Suspensii bune	hl



INSERT INTO

Distribuito(r)(Id\_distribuito(r),Numedis,PrenameDis,Emaildis,TelefonDis,numar\_inmatricul  
are)

Values(1,'Tache','Ion','tacheion@gmail.com','0727544840','B185WHM');

INSERT INTO

Distribuito(r)(Id\_distribuito(r),Numedis,PrenameDis,Emaildis,TelefonDis,numar\_inmatricul  
are)

Values(2,'Ion','Angelo','ionandgelo@gmail.com','0218391124','B06WHM');

INSERT INTO

Distribuito(r)(Id\_distribuito(r),Numedis,PrenameDis,Emaildis,TelefonDis,numar\_inmatricul  
are)

Values(3,'Ion','Irina','ionirina@gmail.com','0729169295','B123FUG');

INSERT INTO

Distribuito(r)(Id\_distribuito(r),Numedis,PrenameDis,Emaildis,TelefonDis,numar\_inmatricul  
are)

Values(4,'Andy','Mihai','andymihai@gmail.com','028338939211','B185WHM');

INSERT INTO

Distribuito(r)(Id\_distribuito(r),Numedis,PrenameDis,Emaildis,TelefonDis,numar\_inmatricul  
are)

Values(5,'Ion','Adina','adionion@gmail.com','02992999229','B01TOP');

INSERT INTO

Distribuito(r)(Id\_distribuito(r),Numedis,PrenameDis,Emaildis,TelefonDis,numar\_inmatricul  
are)

Values(6,'Lungu','Carmen','carmenl@gmail.com','2345670980','B185WHM');

INSERT INTO

Distribuito(r)(Id\_distribuito(r),Numedis,PrenameDis,Emaildis,TelefonDis,numar\_inmatricul  
are)

Values(7,'Pupaza','David','pupazadavid@gmail.com','1234567890','B01TOP');

INSERT INTO

Distribuito(r)(Id\_distribuito(r),Numedis,PrenameDis,Emaildis,TelefonDis,numar\_inmatricul  
are)

Values(8,'Oprea','Daniela','dani@gmail.com','223534647','B06WHM');

## INSERT INTO

Distributor(Id\_distributor,Numedis,PrenumeDis,Emaildis,TelefonDis,numar\_inmatriculare)

Values(9,'Bratu','Carmen','camenbrat@gmail.com','07217319391','B123FUG');

## INSERT INTO

Distributor(Id\_distributor,Numedis,PrenumeDis,Emaildis,TelefonDis,numar\_inmatriculare)

Values(10,'Pupaza','Mioara','pupazamioara@gmail.com','10293627523','B01TOP');

The screenshot shows the SQL Developer interface. On the left, the 'Connections' pane lists the database structure, including the 'DISTRIBUTOR' table. The main window displays the 'Query Result' tab, showing the data inserted into the 'DISTRIBUTOR' table. The data is presented in a table with 7 columns: ID\_DISTRIBUTOR, NUMEDIS, PRENUMEDIS, EMAILDIS, TELEFONDIS, and NUMAR\_I... (partially visible). The table contains 10 rows of data, including the newly inserted records for Bratu and Pupaza.

ID_DISTRIBUTOR	NUMEDIS	PRENUMEDIS	EMAILDIS	TELEFONDIS	NUMAR_I...
1	1 Tache	Ion	tacheion@gmail.com	0727544840	B185WHM
2	2 Ion	Angelo	ionandgelo@gmail.com	0218391124	B06WHM
3	3 Ion	Irina	ionirina@gmail.com	0729169295	B123FUG
4	4 Andy	Mihai	andymihai@gmail.com	028338939211	B185WHM
5	5 Ion	Adina	adionion@gmail.com	02992999229	B01TOP
6	7 Pupaza	David	pupazadavid@gmail.com	1234567890	B01TOP
7	6 Lungu	Carmen	carmenl@gmail.com	2345670980	B185WHM
8	8 Oprea	Daniela	dani@gmail.com	223534647	B06WHM
9	9 Bratu	Carmen	camenbrat@gmail.com	07217319391	B123FUG
10	10 Pupaza	Mioara	pupazamioara@gmail.com	10293627523	B01TOP

```
INSERT INTO
Motociclete_distribuite(numar_inmatriculare,ID_distribuitoar,ObservatiiMD)
VALUES('B185WHM',1,'casca inclusa');
```

```
INSERT INTO
Motociclete_distribuite(numar_inmatriculare,ID_distribuitoar,ObservatiiMD)
VALUES('B123FUG',5,'fara casca inclusa');
```

```
INSERT INTO
Motociclete_distribuite(numar_inmatriculare,ID_distribuitoar,ObservatiiMD)
VALUES('B06WHM',7,'casca inclusa');
```

```
INSERT INTO
Motociclete_distribuite(numar_inmatriculare,ID_distribuitoar,ObservatiiMD)
VALUES('B01TOP',9,'fara casca inclusa');
```

```
INSERT INTO
Motociclete_distribuite(numar_inmatriculare,ID_distribuitoar,ObservatiiMD)
VALUES('B185WHM',2,'suport telefon inclus');
```

```
INSERT INTO
Motociclete_distribuite(numar_inmatriculare,ID_distribuitoar,ObservatiiMD)
VALUES('B185WHM',10,'casca inclusa');
```

```
INSERT INTO
Motociclete_distribuite(numar_inmatriculare,ID_distribuitoar,ObservatiiMD)
VALUES('B123FUG',9,'fara casca inclusa');
```

```
INSERT INTO
Motociclete_distribuite(numar_inmatriculare,ID_distribuitoar,ObservatiiMD)
VALUES('B06WHM',8,'casca inclusa');
```

```
INSERT INTO
Motociclete_distribuite(numar_inmatriculare,ID_distribuitoar,ObservatiiMD)
VALUES('B01TOP',9,'fara casca inclusa');
```

```
INSERT INTO
Motociclete_distribuite(numar_inmatriculare,ID_distribuitoar,ObservatiiMD)
VALUES('B123FUG',2,'suport telefon inclus');
```

The screenshot shows a database management interface with a left sidebar containing a tree view of database objects. The main area is divided into a 'Query Builder' section at the top and a 'Query Result' section at the bottom. The 'Query Builder' section contains two SQL statements: an INSERT statement and a SELECT statement. The 'Query Result' section displays the results of the SELECT statement in a table format.

**Query Builder SQL:**

```
INSERT INTO Motociclete_distribuite (numar_inmatriculare, id_distributor, observatiiMD)
VALUES ('B123FUG', 2, 'suport telefon inclus');

select * from Motociclete_distribuite;
```

**Query Result Table:**

	NUMAR_INMATRICULARE	ID_DISTRIBUTOR	OBSERVATII MD
1	B185WHM		1 casca inclusa
2	B123FUG		5 fara casca inclusa
3	B06WHM		7 casca inclusa
4	B01TOP		9 fara casca inclusa
5	B185WHM		2 suport telefon inclus
6	B185WHM		10 casca inclusa
7	B123FUG		9 fara casca inclusa
8	B06WHM		8 casca inclusa
9	B01TOP		9 fara casca inclusa
10	B123FUG		2 suport telefon inclus

```
INSERT INTO Motociclete_inchiriate (numar_inmatriculare, ID_client, observatiiMI)
VALUES ('B185WHM', 1, 'perioada de 10 zile');
```

```
INSERT INTO Motociclete_inchiriate (numar_inmatriculare, ID_client, observatiiMI)
VALUES ('B06WHM', 2, 'perioada de 30 zile');
```

```
INSERT INTO Motociclete_inchiriate (numar_inmatriculare, ID_client, observatiiMI)
VALUES ('B123FUG', 3, 'perioada de 3 zile');
```

```
INSERT INTO Motociclete_inchiriate (numar_inmatriculare, ID_client, observatiiMI)
VALUES ('B01TOP', 4, 'perioada de 60 zile');
```

```
INSERT INTO Motociclete_inchiriate (numar_inmatriculare, ID_client, observatiiMI)
VALUES ('B06WHM', 5, 'perioada de 107 zile');
```

```
INSERT INTO Motociclete_inchiriate (numar_inmatriculare, ID_client, observatiiMI)
VALUES ('B123FUG', 6, 'perioada de 130 zile');
```

```
INSERT INTO Motociclete_inchiriate (numar_inmatriculare, ID_client, observatiiMI)
VALUES ('B06WHM', 7, 'perioada de 100 zile');
```

```
INSERT INTO Motociclete_inchiriate (numar_inmatriculare, ID_client, observatiiMI)
```

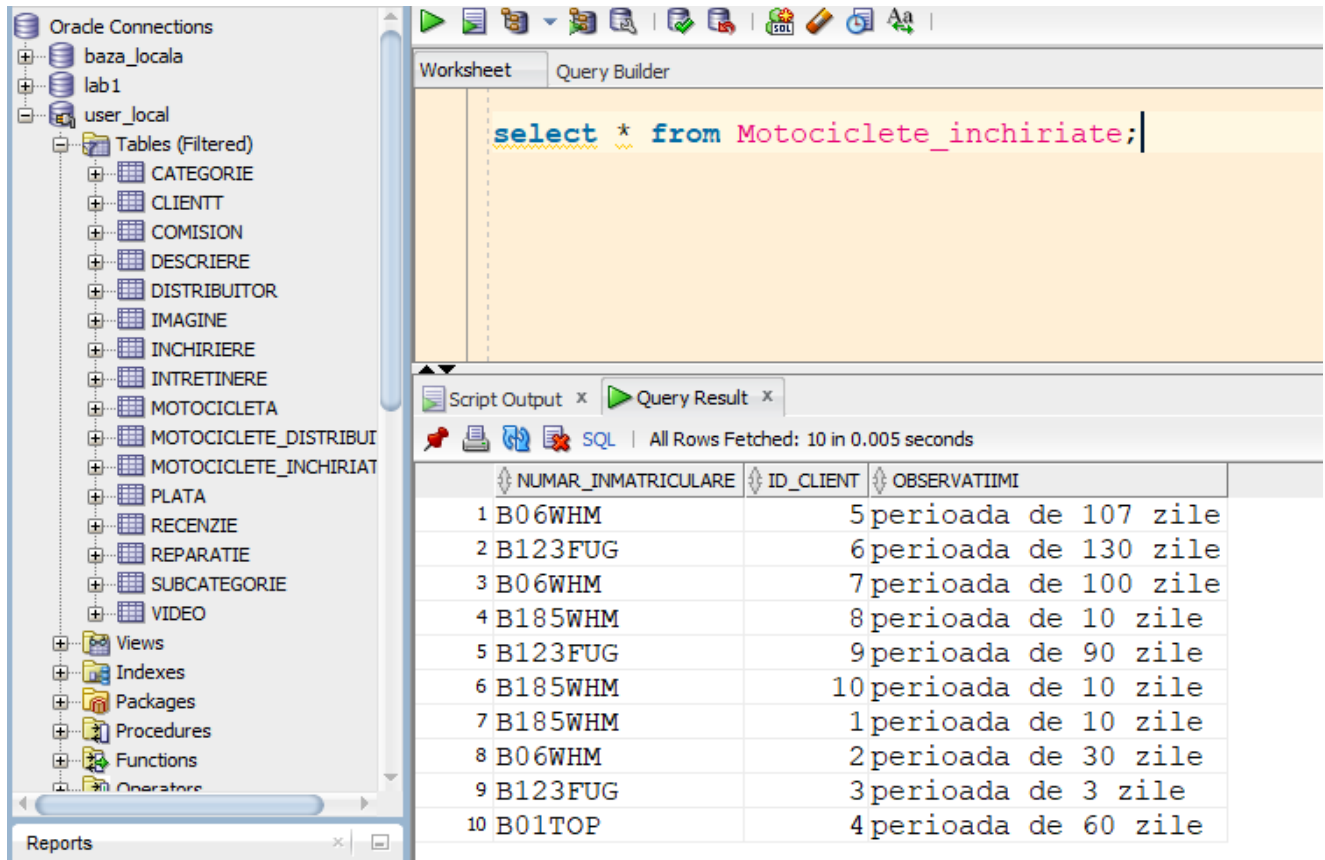
```
VALUES ('B185WHM',8,'perioada de 10 zile');
```

```
INSERT INTO Motociclete_inchiriate(numar_inmatriculare,ID_client,observatiiMI)
```

```
VALUES ('B123FUG',9,'perioada de 90 zile');
```

```
INSERT INTO Motociclete_inchiriate(numar_inmatriculare,ID_client,observatiiMI)
```

```
VALUES ('B185WHM',10,'perioada de 10 zile');
```



The screenshot shows the Oracle SQL Developer interface. On the left, the 'Oracle Connections' tree displays the 'baza\_locala' database. The 'Tables (Filtered)' list includes: CATEGORIE, CLIENTT, COMISION, DESCRIERE, DISTRIBUTOR, IMAGINE, INCHIRIERE, INTRETINERE, MOTOCICLETA, MOTOCICLETE\_DISTRIBUI, MOTOCICLETE\_INCHIRIAT, PLATA, RECENZIE, REPARATIE, SUBCATEGORIE, and VIDEO. The main window shows a 'Query Builder' with the SQL query: `select * from Motociclete_inchiriate;`. Below the query, the 'Query Result' tab displays the results of the query. The status bar indicates 'All Rows Fetched: 10 in 0.005 seconds'.

	NUMAR_INMATRICULARE	ID_CLIENT	OBSERVATIIMI
1	B06WHM	5	perioada de 107 zile
2	B123FUG	6	perioada de 130 zile
3	B06WHM	7	perioada de 100 zile
4	B185WHM	8	perioada de 10 zile
5	B123FUG	9	perioada de 90 zile
6	B185WHM	10	perioada de 10 zile
7	B185WHM	1	perioada de 10 zile
8	B06WHM	2	perioada de 30 zile
9	B123FUG	3	perioada de 3 zile
10	B01TOP	4	perioada de 60 zile

```
INSERT INTO Recenzie(id_recenzie,numar_inmatriculare,id_client,recenzie)
```

```
VALUES(1,'B185WHM',10,'utila pentru transport');
```

```
INSERT INTO Recenzie(id_recenzie,numar_inmatriculare,id_client,recenzie)
```

```
VALUES(2,'B123FUG',9,'prinde foarte repede viteza');
```

```
INSERT INTO Recenzie(id_recenzie,numar_inmatriculare,id_client,recenzie)
```

```
VALUES(3,'B06WHM',7,'superba pentru plimbari');
```

```
INSERT INTO Recenzie(id_recenzie,numar_inmatriculare,id_client,recenzie)
```

```
VALUES(4,'B01TOP',4,'putin cam zgomotoasa');
```

```
INSERT INTO Recenzie(id_recenzie,numar_inmatriculare,id_client,recenzie)
```

```
VALUES(5,'B123FUG',3,'utila pentru transport');
```

The screenshot shows the Oracle SQL Developer interface. On the left, the 'Tables (Filtered)' list includes: CATEGORIE, CLIENTI, COMISION, DESCRIERE, DISTRIBUTOR, IMAGINE, INCHIRIERE, INTRETINERE, MOTOCICLETA, MOTOCICLETE\_DISTRIBUI, MOTOCICLETE\_INCHIRIAT, PLATA, RECENZIE, REPARATIE, SUBCATEGORIE, and VIDEO. The 'RECENZIE' table is selected, showing its columns: ID\_RECENZIE, NUMAR\_INMATRICULARE, ID\_CLIENT, and RECENZIE.

The Query Builder window contains the following SQL script:

```

INSERT INTO Recenzie(id_recenzie,numar_inmatriculare,id_client)
VALUES (4,'B01TOP',4,'putin cam zgomotoasa');
INSERT INTO Recenzie(id_recenzie,numar_inmatriculare,id_client)
VALUES (5,'B123FUG',3,'utila pentru transport');

select * from recenzie;

```

The Query Result window shows the results of the query:

ID_RECENZIE	NUMAR_INMATRICULARE	ID_CLIENT	RECENZIE
1	1B185WHM	10	utila pentru transport
2	2B123FUG	9	prinde foarte repede viteza
3	3B06WHM	7	superba pentru plimbari
4	4B01TOP	4	putin cam zgomotoasa
5	5B123FUG	3	utila pentru transport

```

INSERT INTO Categorie(id_categorie,numecategorie) VALUES(1,'de oras');
INSERT INTO Categorie(id_categorie,numecategorie) VALUES(2,'de munte');
INSERT INTO Categorie(id_categorie,numecategorie) VALUES(3,'de teren lin');
INSERT INTO Categorie(id_categorie,numecategorie) VALUES(4,'de teren accidentat');
INSERT INTO Categorie(id_categorie,numecategorie) VALUES(5,'de teren derapant');

```

The screenshot shows the Oracle SQL Developer interface. The Query Builder window contains the following SQL script:

```

INSERT INTO Categorie(id_categorie,numecategorie) VALUES(1,'de oras');
INSERT INTO Categorie(id_categorie,numecategorie) VALUES(2,'de munte');
INSERT INTO Categorie(id_categorie,numecategorie) VALUES(3,'de teren lin');
INSERT INTO Categorie(id_categorie,numecategorie) VALUES(4,'de teren accidentat');
INSERT INTO Categorie(id_categorie,numecategorie) VALUES(5,'de teren derapant');

select * from categorie;

```

The Query Result window shows the results of the query:

ID_CATEGORIE	NUMECATEGORIE
1	1 de teren
2	2 de oras
3	3 de teren lin
4	4 de teren accidentat
5	5 de teren derapant

```

INSERT INTO Subcategorie(id_subcategorie,numesubcategorie,id_categorie)
VALUES(19,'electrica',1);

INSERT INTO Subcategorie(id_subcategorie,numesubcategorie,id_categorie)
VALUES(17,'benzina',2);

INSERT INTO Subcategorie(id_subcategorie,numesubcategorie,id_categorie)
VALUES(70,'motorina',3);

INSERT INTO Subcategorie(id_subcategorie,numesubcategorie,id_categorie)
VALUES(11,'diesel',4);

INSERT INTO Subcategorie(id_subcategorie,numesubcategorie,id_categorie)
VALUES(15,'otto',5);

```

The screenshot shows the Oracle SQL Developer interface. On the left, the 'Connections' pane shows the 'user\_local' connection. The 'Tables (Filtered)' list includes: CATEGORIE, CLIENTI, COMISION, DESCRIERE, DISTRIBUTOR, IMAGINE, INCHIRIERE, INTRETINERE, MOTOCICLETA, MOTOCICLETE\_DISTRIBUI, MOTOCICLETE\_INCHIRIAT, PLATA, RECENZIE, REPARATIE, SUBCATEGORIE, and VIDEO. The main 'SQL Worksheet' contains the following SQL code:

```

INSERT INTO Subcategorie(id_subcategorie,numesubcategorie,id_categorie)
VALUES(70,'motorina',3);
INSERT INTO Subcategorie(id_subcategorie,numesubcategorie,id_categorie)
VALUES(11,'diesel',4);
INSERT INTO Subcategorie(id_subcategorie,numesubcategorie,id_categorie)
VALUES(15,'otto',5);
select * from subcategorie;

```

Below the worksheet, the 'Query Result' pane shows the output of the SELECT query. It indicates 'All Rows Fetched: 5 in 0.014 seconds' and displays a table with 5 rows and 3 columns: ID\_SUBCATEGORIE, NUMESUBCATEGORIE, and ID\_CATEGORIE.

ID_SUBCATEGORIE	NUMESUBCATEGORIE	ID_CATEGORIE
1	19electrica	1
2	17benzina	2
3	70motorina	3
4	11diesel	4
5	15otto	5

**12.** Formulați în limbaj natural și implementați 5 cereri SQL complexe ce vor utiliza, în ansamblul lor, următoarele elemente:

- subcerere sincronizată și nesincronizată

```
SELECT nume,telefon,data_plata,pret,ObservatiiMi
FROM (Select c.nume,c.telefon,i.data_sfarsit,p.pret,
        i.data_inceput,p.data_plata,i.id_inchiriere,m.ObservatiiMi
FROM Inchiriere I JOIN Clientt c on (c.id_client=i.id_client)
        JOIN plata p on (p.id_inchiriere=i.id_inchiriere)
        JOIN Motociclete_inchiriate m on (m.id_client=c.id_client))
WHERE Data_sfarsit in (Select (data_inceput+id_inchiriere)
FROM Inchiriere);
```

	NUME	TELEFON	DATA_PLATA	PRET	OBSERVATIIMI
1	Argesanu	0778546123	15-JUL-22	320.12	perioada de 60 zile

- grupări de date cu subcereri nesincronizate in care intervin cel puțin 3 tabele, functii grup, filtrare la nivel de grupuri (în cadrul aceleiași cereri)

```
SELECT marca,sum(pret_inchiriere)
FROM (select m.marca,m.pret_inchiriere,d.descriere
      from Motocicleta m join descriere d on
      (m.numar_inmatriculare=d.numar_inmatriculare))
Group by marca having marca='Dacia';
```

	MARCA	SUM(PRET_INCHIRIERE)
1	Dacia	180

- ordonări și utilizarea funcțiilor NVL și DECODE (în cadrul aceleiași cereri)

```
SELECT id_client,nume,prenume, nvl(email,'nu avem mail') email,
      decode(telefon,null,'nu avem telefon',telefon) telefon
FROM Clientt
ORDER BY nume;
```



	ID_CLIENT	NUME	PRENUME	EMAIL	TELEFON
1	10	Andrei	Alina	ali.andrei@hotmail.com	0710935658
2	5	Argesanu	Rodica	argesanu.r@yahoo.com	0778546123
3	9	Constantin	Andrei	crt.r@yahoo.com	0777546123
4	4	Grigore	Alexandru	alex.grigore@gmail.com	0712345999
5	1	Ion	Adina	adinaion53@gmail.com	0799790528
6	7	Ion	Andy	ion.andy@yahoo.com	079931766
7	3	Mihai	Ioana	ioana.mihai@yahoo.com	0799887766
8	11	Mirigel	Radu	nu avem mail	nu avem telefon
9	6	Oprea	Tudor	tudor.oprea@hotmail.com	0710986338
10	2	Popescu	Ion	ion.popescu@gmail.com	0712345678
11	8	Popescu	Mihaela	pop.miha@gmail.com	0717655899

```

SELECT id_client,nume,prenume, nvl(email,'nu avem mail') email,
       decode(email,'ion.andy@yahoo.com','mail secret',email) email_decode
FROM Clientt
ORDER BY prenume;

```

	ID_CLIENT	NUME	PRENUME	EMAIL	EMAIL_DECODE
1	1	Ion	Adina	adinaion53@gmail.com	adinaion53@gmail.com
2	4	Grigore	Alexandru	alex.grigore@gmail.com	alex.grigore@gmail.com
3	10	Andrei	Alina	ali.andrei@hotmail.com	ali.andrei@hotmail.com
4	9	Constantin	Andrei	crt.r@yahoo.com	crt.r@yahoo.com
5	7	Ion	Andy	ion.andy@yahoo.com	mail secret
6	3	Mihai	Ioana	ioana.mihai@yahoo.com	ioana.mihai@yahoo.com
7	2	Popescu	Ion	ion.popescu@gmail.com	ion.popescu@gmail.com
8	8	Popescu	Mihaela	pop.miha@gmail.com	pop.miha@gmail.com
9	11	Mirigel	Radu	nu avem mail	(null)
10	5	Argesanu	Rodica	argesanu.r@yahoo.com	argesanu.r@yahoo.com
11	6	Oprea	Tudor	tudor.oprea@hotmail.com	tudor.oprea@hotmail.com

•utilizarea a cel puțin 2 funcții pe șiruri de caractere, 2 funcții pe date calendaristice, a cel puțin unei expresii CASE

```


SELECT id_intretinere, numar_inmatriculare,data,cost ,
CASE WHEN numar_inmatriculare='B999wee' THEN UPPER(numar_inmatriculare)
      WHEN numar_inmatriculare='B123 NO' THEN
REPLACE(numar_inmatriculare,numar_inmatriculare,'B123NO')
ELSE numar_inmatriculare
END numere_noi,

```

```

CASE WHEN data=TO_DATE('2015-07-15', 'YYYY-MM-DD') THEN ADD_MONTHS(data,5)
      WHEN data=TO_DATE('2022-05-10', 'YYYY-MM-DD') THEN ROUND(data,'yyyy')
ELSE data
END date_noi
FROM Intretinere;

```

 All Rows Fetched: 12 in 0.005 seconds

	ID_INTRETINERE	NUMAR_INMATRICULARE	DATA	COST	NUMERE_NOI	DATE_NOI
1		1 B185WHM	20-DEC-22	100	B185WHM	20-DEC-22
2		2 B185WHM	12-SEP-22	250	B185WHM	12-SEP-22
3		3 B06WHM	01-OCT-22	150	B06WHM	01-OCT-22
4		4 B123FUG	25-NOV-22	25	B123FUG	25-NOV-22
5		5 B185WHM	20-MAY-22	120	B185WHM	20-MAY-22
6		6 B07WHM	20-DEC-22	100	B07WHM	20-DEC-22
7		7 B01TOP	20-DEC-22	190	B01TOP	20-DEC-22
8		8 B185WHM	10-MAY-22	300	B185WHM	01-JAN-22
9		9 B123FUG	05-JAN-22	70	B123FUG	05-JAN-22
10		10 B06WHM	15-JAN-22	30	B06WHM	15-JAN-22
11		12 B123 NO	15-JUL-22	999	B123NO	15-JUL-22
12		13 B999wee	15-JUL-15	1234	B999WEE	15-DEC-15

•utilizarea a cel puțin 1 bloc de cerere (clauza WITH)

```

with numere_rows as (select ROW_NUMBER() OVER (ORDER BY id_intretinere) AS
row_number, id_intretinere, numar_inmatriculare, data
from intretinere)
SELECT id_intretinere, numar_inmatriculare,data,
CASE WHEN numar_inmatriculare='B999wee' THEN UPPER(numar_inmatriculare)
      WHEN numar_inmatriculare='B123 NO' THEN
REPLACE(numar_inmatriculare,numar_inmatriculare,'B123NO')
ELSE numar_inmatriculare
END numere_noi,

```

```

CASE WHEN data=TO_DATE('2015-07-15', 'YYYY-MM-DD') THEN ADD_MONTHS(data,5)
      WHEN data=TO_DATE('2022-05-10', 'YYYY-MM-DD') THEN ROUND(data,'yyyy')
ELSE data
END date_noi
FROM numere_rows where row_number>3;

```

	ID_INTRETINERE	NUMAR_INMATRICULARE	DATA	NUMERE_NOI	DATE_NOI
1	4	B123FUG	25-NOV-22	B123FUG	25-NOV-22
2	5	B185WHM	20-MAY-22	B185WHM	20-MAY-22
3	6	B07WHM	20-DEC-22	B07WHM	20-DEC-22
4	7	B01TOP	20-DEC-22	B01TOP	20-DEC-22
5	8	B185WHM	10-MAY-22	B185WHM	01-JAN-22
6	9	B123FUG	05-JAN-22	B123FUG	05-JAN-22
7	10	B06WHM	15-JAN-22	B06WHM	15-JAN-22
8	12	B123 NO	15-JUL-22	B123NO	15-JUL-22
9	13	B999wee	15-JUL-15	B999WEE	15-DEC-15

**13.** Implementarea a 3 operații de actualizare și de suprimare a datelor utilizând subcereri.

-- stergerea motocicletelor care nu au imagine

```

delete from Motocicleta where numar_inmatriculare not in (select numar_inmatriculare
FROM Imagine);

```

rollback;

-- creșterea cu 10% a prețului motocicletei închiriată de

-- clientul cu id-ul 4

```

select * from plata;

```

```

update Plata set pret=pret*1.1

```

```

where id_client in (select id_client from Clientt where id_client=4);

```

rollback;

-- ștergerea descrierilor motocicletelor care au Videoclip

```

delete from descriere where numar_inmatriculare in (select numar_inmatriculare from
Video);

```

rollback;

-- cele care au 'B185WHM' devin cu statutul observațiilor de schimbat

```

update Motociclete_inchiriate set observatiiMI='schimbat'

```

```
where numar_inmatriculare in (select numar_inmatriculare from Motocicleta where
numar_inmatriculare='B185WHM');
```

```
rollback;
```

-- ștergerea clienților care nu au închiriat motociclete

```
delete from Clientt where id_client not in(select id_client from Motociclete_inchiriate);
```

```
rollback;
```

-- clienții care au dat recenzii

```
update Clientt set prenume='recenzie pe MI'
```

```
where id_client in (select id_client from Recenzie);
```

```
select * from recenzie;
```

```
select * from clientt;
```

**14.** Crearea unei vizualizări complexe. Dați un exemplu de operație LMD permisă pe vizualizarea respectivă și un exemplu de operație LMD nepermisă.

```
CREATE VIEW Moto_Recenzie
```

```
AS SELECT m.numar_inmatriculare numar_inmatriculare,marca,an_fabricatie,recenzie
```

```
FROM Motocicleta m join Recenzie r on m.numar_inmatriculare=r.numar_inmatriculare;
```

```
select * from Moto_recenzie;
```

```
CREATE VIEW Moto_Recenzie
AS SELECT m.numar_inmatriculare numar_inmatriculare,marca,an_fabricatie,recenzie
FROM Motocicleta m join Recenzie r on m.numar_inmatriculare=r.numar_inmatriculare;
select * from Moto_recenzie;
```

ipt Output x Query Result x Query Result 1 x Query Result 2 x

SQL | All Rows Fetched: 5 in 0.01 seconds

NUMAR_INMATRICULARE	MARCA	AN_FABRICATIE	RECENZIE
1 B01TOP	BMW	2010	putin cam zgomotoasa
2 B06WHM	Dacia	2001	superba pentru plimbări
3 B123FUG	Volkswagen	2013	utila pentru transport
4 B123FUG	Volkswagen	2013	prinde foarte repede viteza
5 B185WHM	Volkswagen	2018	utila pentru transport

-- LMD permisă

```
update Moto_recenzie set marca='Audi' where numar_inmatriculare='B06WHM';
```

```

CREATE VIEW Moto_Recenzie
AS SELECT distinct m.numar_inmatriculare
numar_inmatriculare,marca,an_fabricatie,recenzie
FROM Motocicleta m join Recenzie r on m.numar_inmatriculare=r.numar_inmatriculare;
select * from Moto_recenzie;
--LMD nepermisa
update Moto_recenzie set marca='Audi' where numar_inmatriculare='B06WHM';

```

*Error report -*

*SQL Error: ORA-01732: data manipulation operation not legal on this view*

*01732. 00000 - "data manipulation operation not legal on this view"*

**15.** Formulați în limbaj natural și implementați în SQL: o cerere ce utilizează operația outer-join pe minimum 4 tabele, o cerere care utilizează operația division și o cerere care implementează analiza top-n. Observație: Cele 3 cereri sunt diferite de cererile de la exercițiul 12.

-- outer join pe minim 4 tabele

-- afișez numărul de înmatriculare motocicletelor care au

-- recenzie, imagine, video, descriere

```

SELECT numar_inmatriculare
FROM (SELECT 'Imagine' name,numar_inmatriculare
      FROM Imagine
      UNION ALL
      SELECT 'Video' name,numar_inmatriculare
      FROM Video
      UNION ALL
      SELECT 'Descriere' name,numar_inmatriculare
      FROM Descriere
      UNION ALL

```

```

SELECT 'Recenzie' name,numar_inmatriculare
FROM Recenzie) ni
GROUP BY numar_inmatriculare;

```

	NUMAR_INMATRICULARE
1	B185WHM
2	B06WHM
3	B123FUG
4	B01TOP
5	B07WHM

-- division

```

SELECT trunc(cost / id_reparatie)
FROM Reparatie where id_reparatie=3 or id_reparatie=1;

```

	TRUNC(COST/ID_REPARATIE)
1	200
2	116

-- analiza Top-N

```

SELECT id_intretinere, numar_inmatriculare,data,cost
FROM (SELECT id_intretinere, numar_inmatriculare,data,cost
      FROM Intretinere
      ORDER BY cost)
WHERE cost>=100;

```

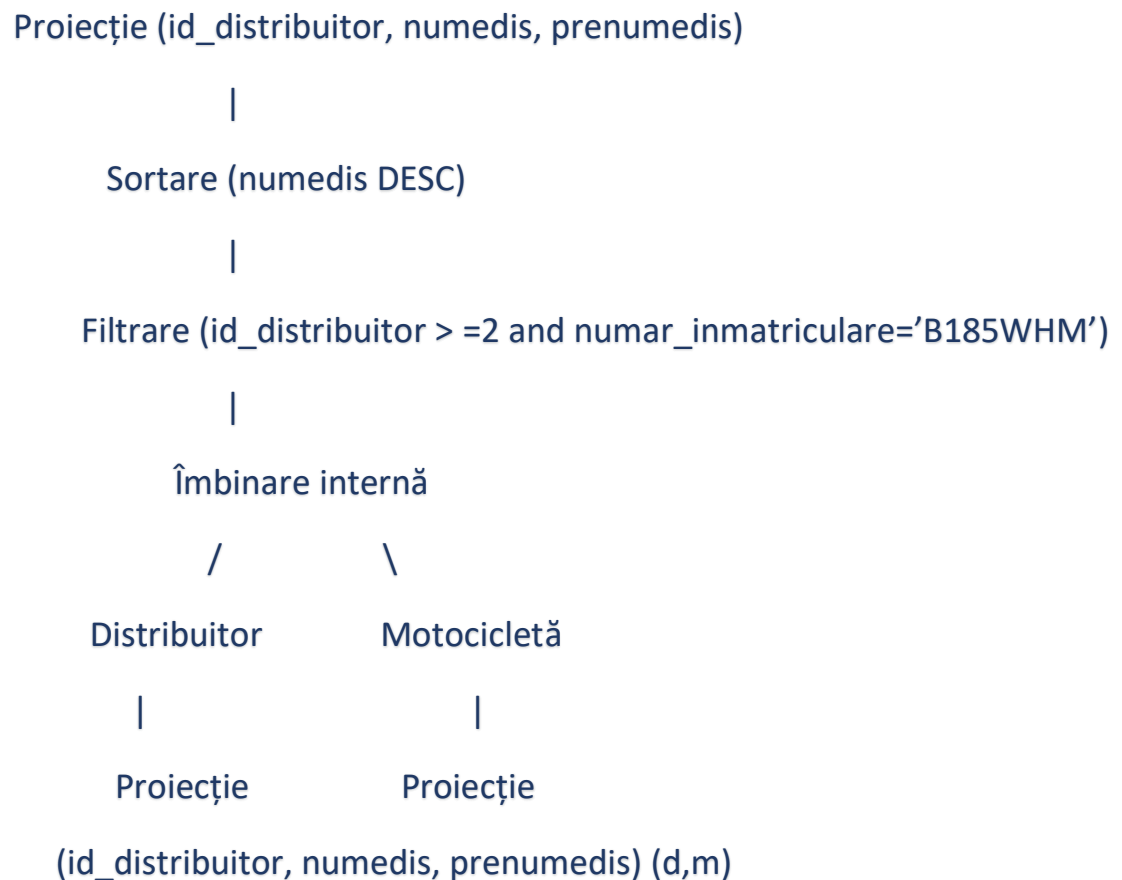
	ID_INTRETINERE	NUMAR_INMATRICULARE	DATA	COST
1	1	B185WHM	20-DEC-22	100
2	6	B07WHM	20-DEC-22	100
3	5	B185WHM	20-MAY-22	120
4	3	B06WHM	01-OCT-22	150
5	7	B01TOP	20-DEC-22	190
6	2	B185WHM	12-SEP-22	250
7	8	B185WHM	10-MAY-22	300

**16.** Optimizarea unei cereri, aplicând regulile de optimizare ce derivă din proprietățile operatorilor algebrei relaționale. Cererea va fi exprimată prin expresie algebrică, arbore algebric și limbaj (SQL), atât anterior cât și ulterior optimizării.

*Iau această cerere:*

```
SELECT d.id_distribuitoar, d.numedis, d.prenumedis
FROM distribuitor d
INNER JOIN motocicletă m ON d.numar_inmatriculare = m.numar_inmatriculare
WHERE d.id_distribuitoar >= 2 AND d.numar_inmatriculare = 'B185WHM'
ORDER BY d.numedis DESC;
```

a) construirea arborelui algebric corespunzător

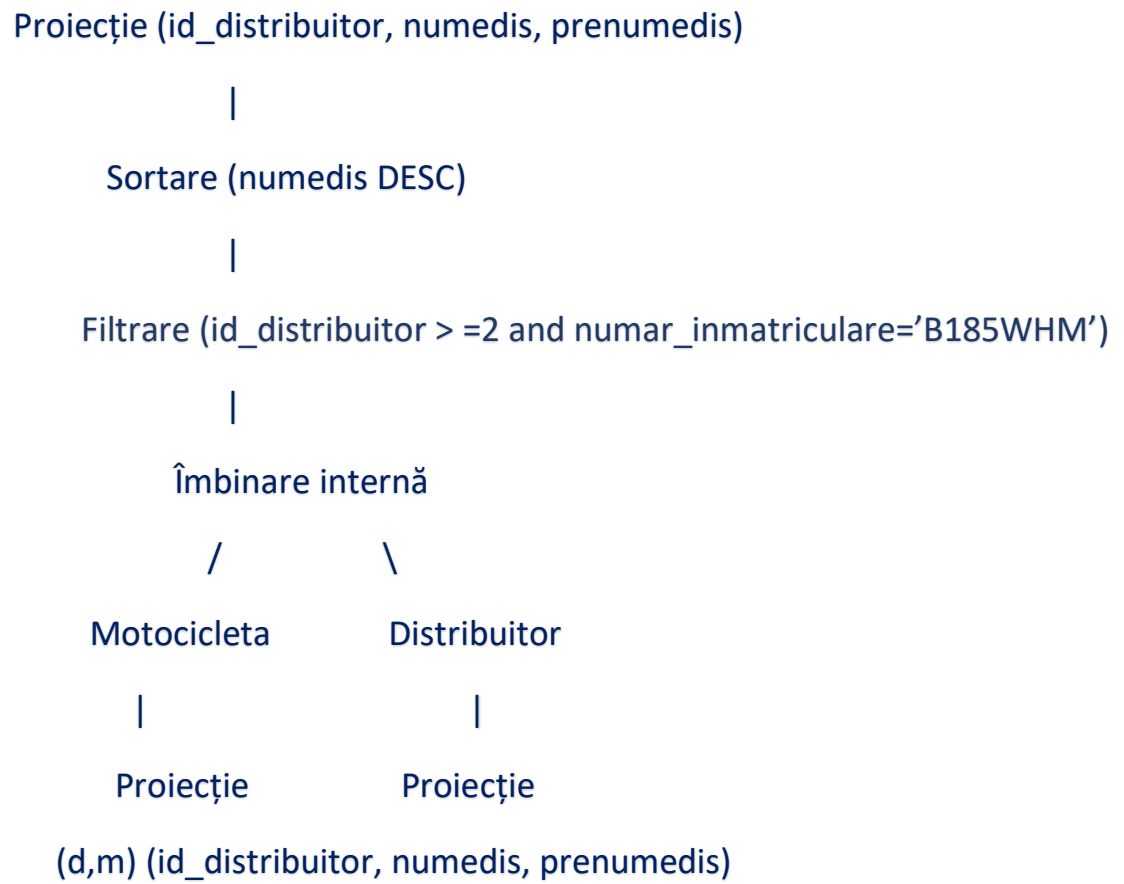


b) Aplicarea regulilor de optimizare:

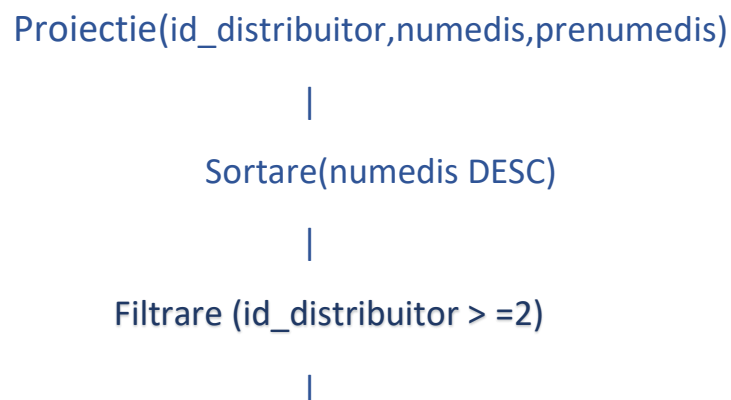
1. Comutativitatea: ordinea imbinărilor poate fi schimbată fără a afecta rezultatul final.

Îmbin datele înainte de a aplica filtrele. Schimb ordinea imbinării și aduc tabelul

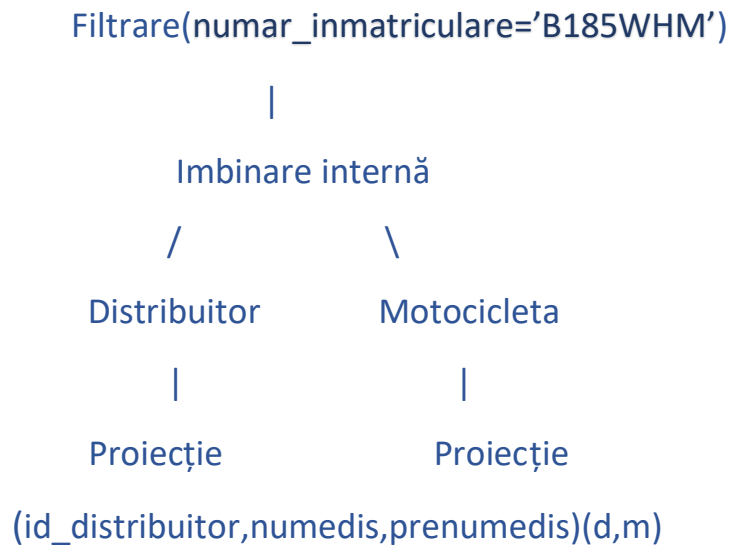
*Motocicleta* înaintea tabelului *Distribuitoar*



2. Distributivitatea: în cazul în care 2 sau mai multe filtre sunt combinate cu un Operator logic, putem să le distribuim în interiorul arborelui algebric pentru a Reduce complexitatea cererii.







c) transpunerea expresiei algebrice optimizate într-o interogare SQL:

```
SELECT d.id_distribuitoar, d.numedis, d.prenumedis
FROM (
    SELECT id_distribuitoar, numedis, prenumedis, numar_inmatriculare
    FROM distribuitoar
    WHERE id_distribuitoar >= 2
) d
INNER JOIN (
    SELECT numar_inmatriculare
    FROM Motocicleta
    WHERE numar_inmatriculare = 'B185WHM'
) m ON d.numar_inmatriculare = m.numar_inmatriculare
ORDER BY d.numedis DESC;
```

	ID_DISTRIBUTOR	NUMEDIS	PRENUMEDIS
1	6	Lungu	Carmen
2	4	Andy	Mihai

## 17.

### a. Realizarea normalizării BCNF, FN4, FN5

#### tabelul in FN3:

- ❖ Motocicleta (numar\_inmatriculare#,marca,an\_fabricatie,pret\_inchiriere, Id\_subcategorie)
- ❖ Reparatie (id\_reparatie#,numar\_inmatriculare#,cost,data)
- ❖ Recenzie (id\_recenzie#,numar\_inmatriculare,recenzie,id\_client)
- ❖ Client (id\_client#,nume,prenume,telefon,email, motociclete\_inchiriate)
- ❖ Motociclete\_inchiriate (numar\_inmatriculare#,id\_client#,observatiiMi)
- ❖ Inchiriere(id\_inchiriere#,id\_client,numar\_inmatriculare,data\_inceput,data\_sfarsit,pret\_total)
- ❖ Categori (categori\_id#,numeCategorie)
- ❖ Comision (id\_comision#,id\_reparatie,procent\_comision)
- ❖ Motociclete\_distribuite(numar\_inmatriculare#,observatiiMD,distribuitoar\_id#)
- ❖ Descriere (id\_descriere#,numar\_inmatriculare,descriere,format\_descriere)
- ❖ Plata (id\_plata#,id\_client,id\_inchiriere,data\_plata,pret)
- ❖ Distribuitoar (distribuitoar\_id,numeDis,prenumeDis,telefonDis,emailDis,oras)
- ❖ Video (id\_video#,numar\_inmatriculare,numeVideo,formatAudio,video)
- ❖ Imagine (id\_img#,numar\_inmatriculare,rezolutie,nume\_img,tip\_img)
- ❖ Intretinere (id\_intretinere,numar\_inmatriculare,data,cost)
- ❖ Subcategorie (Id\_subcategorie#,numeSubcategorie,Id\_categorie)

#### -- realizarea normalizării FN4

Realizarea descompunerilor necesare pentru a atinge forma normală FN4: Descompunem tabelele care conțin dependențe funcționale tranzitive în tabele separate. În acest caz, putem observa că tabela *Motocicleta* conține o dependență funcțională tranzitivă prin intermediul atributului Id\_subcategorie. Vom descompune tabela *Motocicleta* în două tabele separate: *Motocicleta* și *Subcategorie*, eliminând astfel dependența funcțională tranzitivă.

De asemenea, vom descompune tabela *Client* pentru a elimina dependența funcțională multiatribut din atributul motociclete\_inchiriate.

#### Tabele rezultate în urma descompunerilor:

- ❖ Motocicleta (numar\_inmatriculare#, marca, an\_fabricatie, pret\_inchiriere)
- ❖ Subcategorie (Id\_subcategorie#, numeSubcategorie, Id\_categorie)

- ❖ Reparatie (id\_reparatie#, numar\_inmatriculare#, cost, data)
- ❖ Recenzie (id\_recenzie#, numar\_inmatriculare, recenzie, id\_client)
- ❖ Client (id\_client#, nume, prenume, telefon, email)
- ❖ Motociclete\_inchiriate (numar\_inmatriculare#, id\_client#, observatiiMi)
- ❖ Inchiriere (id\_inchiriere#, id\_client, numar\_inmatriculare, data\_inceput, data\_sfarsit, pret\_total)
- ❖ Categori (categori\_id#, numeCategorie)
- ❖ Comision (id\_comision#, id\_reparatie, procent\_comision)
- ❖ Motociclete\_distribuite (numar\_inmatriculare#, observatiiMD, distribuitor\_id#)
- ❖ Descriere (id\_descriere#, numar\_inmatriculare, descriere, format\_descriere)
- ❖ Plata (id\_plata#, id\_client, id\_inchiriere, data\_plata, pret)
- ❖ Distribuitor (distribuitor\_id, numeDis, prenumeDis, telefonDis, emailDis, oras)
- ❖ Video (id\_video#, numar\_inmatriculare, numeVideo, formatAudio, video)
- ❖ Imagine (id\_img#, numar\_inmatriculare, rezolutie, nume\_img, tip\_img)
- ❖ Întreținere (id\_intretinere, numar\_inmatriculare, data, cost)

Această descompunere asigură că fiecare tabel respectă forma normală FN4, eliminând dependențele funcționale tranzitive și dependențele funcționale multiatribut.

#### -- realizarea normalizării FN5

Nu am dependențe funcționale complexe în schema de mai sus. Există doar dependențe funcționale simple între cheile primare și atributele non-cheie în tabelele din cadrul bazei mele de date.

Nu există dependențe funcționale multivaluate sau dependențe funcționale determinate tranzitiv între atributele cheie primară și celelalte atribute în cadrul tabelor.

Deoarece nu există dependențe funcționale complexe, dependențe funcționale multivaluate sau dependențe funcționale determinate tranzitiv în cadrul schemei date, pot afirma ca schema e deja în FN5.

Astfel, schema prezentată este deja în FN5 și nu este necesară nicio modificare suplimentară.

b. Aplicarea denormalizării, justificând necesitatea acesteia.

În baza mea de date, am mai multe tabele care conțin informații despre motociclete, închirieri, reparații etc. Ca să fac o denormalizare, pot să combin într-un singur tabel informațiile relevante cu detalii.

De exemplu, pot să am doar un tabel "InchirieriDetaliate" care să includă informații despre închirieri, motociclete, clienți și alte detalii relevante. Aceasta ar putea conține coloane precum "id\_inchiriere", "numar\_inmatriculare", "marca", "an\_fabricatie", "pret\_inchiriere", "nume", "prenume", "telefon", "email", "data\_inceput", "data\_sfarsit", "pret\_total" etc. Astfel, toate informațiile necesare pentru o închiriere ar fi stocate într-un singur rând din această tabelă denormalizată.

Denormalizarea în acest caz ar ajuta la simplificarea interogărilor care implică informații despre închirieri, clienți și motociclete, evitând necesitatea îmbinării multiple și oferind o performanță mai bună în ceea ce privește accesul la date.

Așadar, ca răspuns în ceea ce privește necesitatea am: îmbunătățirea performanței, reducerea complexității interogărilor și simplificarea structurii datelor.

