

# TransportCompany-in-mysql

În acest proiect, am dezvoltat o bază de date relațională pentru gestionarea activităților unei companii de transport, care include gestionarea **șoferilor, camioanelor, rutelor, clienților și comenzilor**. Proiectul a acoperit toate etapele esențiale ale dezvoltării bazei de date, de la definirea tabelelor și a relațiilor dintre acestea, până la manipularea și interogarea datelor.

## 1. Crearea și Popularea Tabelelor

Am creat cinci tabele esențiale (**Soferi, Camioane, Rute, Clienti, Comenzi**), fiecare cu coloane specifice pentru a stoca informații detaliate. Aceste tabele au fost populate cu date inițiale pentru a permite testarea funcționalităților.

## 2. Definirea Cheilor Primare și Străine

Am definit chei primare pentru a asigura unicitatea fiecărui rând din tabelă și chei străine pentru a menține relațiile între tabele. Acest lucru a permis integritatea referențială și a facilitat realizarea de interogări complexe.

## 3. Gestionarea Constrângerilor și Integrității Datelor

Am utilizat **ALTER TABLE** pentru a modifica structura tabelelor, **DROP** pentru a elimina tabele sau coloane și **TRUNCATE** pentru a goli tabelele. Am învățat cum să gestionez erorile de constrângere și să mențin integritatea datelor.

## 4. Revenirea la Starea Inițială

Am abordat metodele de revenire la starea inițială a datelor după diverse operațiuni de manipulare, inclusiv ștergerea și reinserarea datelor sau actualizarea constrângerilor.

## 5. Manipularea Datelor cu DML

Am folosit instrucțiuni DML (**INSERT, DELETE, UPDATE, SELECT**) pentru a adăuga, elimina, modifica și selecta date din tabele. Aceasta a permis să se gestioneze în mod eficient datele din baza de date.

## 6. Interogarea Datelor cu DQL

Am utilizat diverse tehnici de interogare:

- **SELECT \*** pentru a obține toate datele din tabele.
- **SELECT** pe coloane specifice pentru a obține informații detaliate.
- Filtrări cu **WHERE, LIKE, AND, OR, NOT** pentru a extrage date relevante.
- Funcții agregate (**AVG, SUM, COUNT, MAX, MIN**) pentru a realiza analize sumative.
- Grupare și filtrare cu **GROUP BY** și **HAVING**.

## 7. Joinuri între Tabele

Am implementat diverse tipuri de joinuri (**INNER JOIN, LEFT JOIN, RIGHT JOIN, CROSS JOIN**) pentru a combina date din mai multe tabele și a obține perspective complete și interconectate asupra datelor.

# I. Crearea și Popularea Tabelelor - Instrucțiuni DDL (Data Definition Language)

- Crearea Bazei de Date

## 1. Tabelul pentru Șoferi

```
-- Crearea tabelului
CREATE TABLE Soferi (
    SoferID INT PRIMARY KEY AUTO_INCREMENT,
    Nume VARCHAR(50),
    Prenume VARCHAR(50),
    DataAngajarii DATE,
    Salariu DECIMAL(10, 2)
);
```

## 2. Tabelul pentru Camioane

```
CREATE TABLE Camioane (
    CamionID INT PRIMARY KEY AUTO_INCREMENT,
    NumarInmatriculare VARCHAR(20),
    Marca VARCHAR(50),
    Model VARCHAR(50),
    AnFabricatie INT,
    CapacitateTonaj DECIMAL(5, 2)
);
```

## 3. Tabelul pentru Rute

```
CREATE TABLE Rute (
    RutaID INT PRIMARY KEY AUTO_INCREMENT,
    Descriere VARCHAR(255),
    DistantaKM DECIMAL(10, 2)
);
```

## 4. Tabelul pentru Clienți

```
CREATE TABLE Clienti (
    ClientID INT PRIMARY KEY AUTO_INCREMENT,
    Nume VARCHAR(50),
    Prenume VARCHAR(50),
    Adresa VARCHAR(100),
    Telefon VARCHAR(20)
);
```

## 5. Tabelul pentru Comenzi

```
CREATE TABLE Comenzi (
    ComandaID INT PRIMARY KEY AUTO_INCREMENT,
    ClientID INT,
    SoferID INT,
    CamionID INT,
    RutaID INT,
    DataComanda DATE,
    Status VARCHAR(20),
    FOREIGN KEY (ClientID) REFERENCES Clienti(ClientID),
    FOREIGN KEY (SoferID) REFERENCES Soferi(SoferID),
    FOREIGN KEY (CamionID) REFERENCES Camioane(CamionID),
    FOREIGN KEY (RutaID) REFERENCES Rute(RutaID)
);
```

Chei Primare și Chei Secundare

- Cheia primară (**PRIMARY KEY**) asigură unicitatea fiecărui rând în tabelă.
- Cheia secundară (**FOREIGN KEY**) menține relațiile dintre tabele.

## ● Inserarea datelor

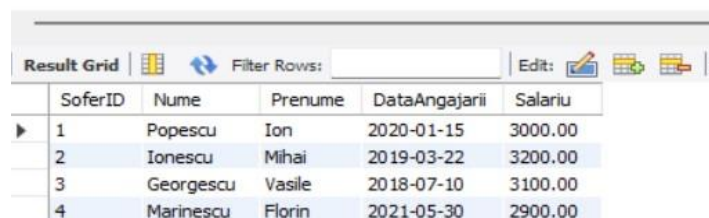
Aceste instrucțiuni de inserare adaugă date în fiecare tabel în conformitate cu relațiile definite. De exemplu, comanda va fi atribuită unui client, va folosi o anumită rută, va fi efectuată de un anumit șofer și va utiliza un anumit camion.

```
-- Inserarea datelor în tabela Soferi
INSERT INTO Soferi (Nume, Prenume, DataAngajarii, Salariu) VALUES
('Popescu', 'Ion', '2020-01-15', 3000.00),
('Ionescu', 'Mihai', '2019-03-22', 3200.00),
('Georgescu', 'Vasile', '2018-07-10', 3100.00),
('Marinescu', 'Florin', '2021-05-30', 2900.00);
```

```
8 ● CREATE TABLE Soferi (
9     SoferID INT PRIMARY KEY AUTO_INCREMENT,
10    Nume VARCHAR(50),
11    Prenume VARCHAR(50),
12    DataAngajarii DATE,
13    Salariu DECIMAL(10, 2)
14 );
```

```
94 -- Inserarea datelor în tabela Soferi
95 ● INSERT INTO Soferi (Nume, Prenume, DataAngajarii, Salariu) VALUES
96 ('Popescu', 'Ion', '2020-01-15', 3000.00),
97 ('Ionescu', 'Mihai', '2019-03-22', 3200.00),
98 ('Georgescu', 'Vasile', '2018-07-10', 3100.00),
99 ('Marinescu', 'Florin', '2021-05-30', 2900.00);
```

```
129 -- Selectarea tuturor datelor din tabela Soferi
130 ● SELECT * FROM Soferi;
```



	SoferID	Nume	Prenume	DataAngajarii	Salariu
▶	1	Popescu	Ion	2020-01-15	3000.00
	2	Ionescu	Mihai	2019-03-22	3200.00
	3	Georgescu	Vasile	2018-07-10	3100.00
	4	Marinescu	Florin	2021-05-30	2900.00

```
-- Inserarea datelor în tabela Camioane
INSERT INTO Camioane (NumarInmatriculare, Marca, Model, AnFabricatie, CapacitateTonaj) VAL
('B123ABC', 'Mercedes', 'Actros', 2015, 20.5),
('B234BCD', 'Volvo', 'FH16', 2018, 22.0),
('B345CDE', 'Scania', 'R450', 2017, 19.0),
('B456DEF', 'DAF', 'XF105', 2019, 21.5);
```

```

16 • CREATE TABLE Camioane (
17     CamionID INT PRIMARY KEY AUTO_INCREMENT,
18     NumarInmatriculare VARCHAR(20),
19     Marca VARCHAR(50),
20     Model VARCHAR(50),
21     AnFabricatie INT,
22     CapacitateTonaj DECIMAL(5, 2)
23 );

```

```







101 -- Inserarea datelor în tabela Camioane
102 • INSERT INTO Camioane (NumarInmatriculare, Marca, Model, AnFabricatie, CapacitateTonaj) VALUES
103 ('B123ABC', 'Mercedes', 'Actros', 2015, 20.5),
104 ('B234BCD', 'Volvo', 'FH16', 2018, 22.0),
105 ('B345CDE', 'Scania', 'R450', 2017, 19.0),
106 ('B456DEF', 'DAF', 'XF105', 2019, 21.5);

```

```

132 -- Selectarea tuturor datelor din tabela Camioane
133 • SELECT * FROM Camioane;

```

Result Grid						
Filter Rows: <input type="text"/>						
Edit:    						
Export/Import:  						
	CamionID	NumarInmatriculare	Marca	Model	AnFabricatie	CapacitateTonaj
▶	1	B123ABC	Mercedes	Actros	2015	20.50
	2	B234BCD	Volvo	FH16	2018	22.00
	3	B345CDE	Scania	R450	2017	19.00
	4	B456DEF	DAF	XF105	2019	21.50

```

-- Inserarea datelor în tabela Rute
INSERT INTO Rute (Descriere, DistanțaKM) VALUES
('București - Constanța', 225.5),
('București - Cluj', 450.0),
('București - Timisoara', 530.0),
('București - Iasi', 400.0);

```

```

25 • CREATE TABLE Rute (
26     RutaID INT PRIMARY KEY AUTO_INCREMENT,
27     Descriere VARCHAR(255),
28     DistanțaKM DECIMAL(10, 2)
29 );

```

```





108 -- Inserarea datelor în tabela Rute
109 • INSERT INTO Rute (Descriere, DistanțaKM) VALUES
110 ('București - Constanța', 225.5),
111 ('București - Cluj', 450.0),
112 ('București - Timisoara', 530.0),
113 ('București - Iasi', 400.0);

```

```

138 -- Selectarea tuturor datelor din tabela Rute
139 • SELECT * FROM Rute;

```

Result Grid		
Filter Rows: <input type="text"/>		
Edit:    		
	RutaID	Descriere
▶	1	București - Constanța
	2	București - Cluj
	3	București - Timisoara
	4	București - Iasi

```
-- Inserarea datelor în tabela Clienti
INSERT INTO Clienti (Nume, Prenume, Adresa, Telefon) VALUES
('Popa', 'Andrei', 'Strada Libertatii 10, Bucuresti', '0722000000'),
('Dumitrescu', 'Ana', 'Strada Independentei 20, Constanta', '0722111111'),
('Stan', 'Maria', 'Strada Victoriei 30, Cluj', '0722222222'),
('Nicolae', 'George', 'Strada Unirii 40, Timisoara', '0722333333');
```

```
31 • CREATE TABLE Clienti (
32     ClientID INT PRIMARY KEY AUTO_INCREMENT,
33     Nume VARCHAR(50),
34     Prenume VARCHAR(50),
35     Adresa VARCHAR(100),
36     Telefon VARCHAR(20)
37 );
```

```
115 -- Inserarea datelor în tabela Clienti
116 • INSERT INTO Clienti (Nume, Prenume, Adresa, Telefon) VALUES
117 ('Popa', 'Andrei', 'Strada Libertatii 10, Bucuresti', '0722000000'),
118 ('Dumitrescu', 'Ana', 'Strada Independentei 20, Constanta', '0722111111'),
119 ('Stan', 'Maria', 'Strada Victoriei 30, Cluj', '0722222222'),
120 ('Nicolae', 'George', 'Strada Unirii 40, Timisoara', '0722333333');
```

```
135 -- Selectarea tuturor datelor din tabela Clienti
136 • SELECT * FROM Clienti;
```

Result Grid					
Filter Rows:					
Edit: Export/Import:					
	ClientID	Nume	Prenume	Adresa	Telefon
▶	1	Popa	Andrei	Strada Libertatii 10, Bucuresti	0722000000
	2	Dumitrescu	Ana	Strada Independentei 20, Constanta	0722111111
	3	Stan	Maria	Strada Victoriei 30, Cluj	0722222222
	4	Nicolae	George	Strada Unirii 40, Timisoara	0722333333

```
-- Inserarea datelor în tabela Comenzi
INSERT INTO Comenzi (ClientID, SoferID, CamionID, RutaID, DataComanda, Status) VALUES
(1, 1, 1, 1, '2023-06-01', 'Livrat'),
(2, 2, 2, 2, '2023-06-15', 'In Tranzit'),
(3, 3, 3, 3, '2023-07-01', 'Livrat'),
(4, 4, 4, 4, '2023-07-15', 'Anulat');
```

```
39 • CREATE TABLE Comenzi (
40     ComandaID INT PRIMARY KEY AUTO_INCREMENT,
41     ClientID INT,
42     SoferID INT,
43     CamionID INT,
44     RutaID INT,
45     DataComanda DATE,
46     Status VARCHAR(20),
47     FOREIGN KEY (ClientID) REFERENCES Clienti(ClientID),
48     FOREIGN KEY (SoferID) REFERENCES Soferi(SoferID),
49     FOREIGN KEY (CamionID) REFERENCES Camioane(CamionID),
50     FOREIGN KEY (RutaID) REFERENCES Rute(RutaID)
51 );
```

```

122 -- Inserarea datelor în tabela Comenzi
123 • INSERT INTO Comenzi (ClientID, SoferID, CamionID, RutaID, DataComanda, Status) VALUES
124 (1, 1, 1, 1, '2023-06-01', 'Livrat'),
125 (2, 2, 2, 2, '2023-06-15', 'In Tranzit'),
126 (3, 3, 3, 3, '2023-07-01', 'Livrat'),
127 (4, 4, 4, 4, '2023-07-15', 'Anulat');

```

```

141 -- Selectarea tuturor datelor din tabela Comenzi
142 • SELECT * FROM Comenzi;

```

Result Grid							
Filter Rows:							
Edit:							
Export/Import:							
ComandaID	ClientID	SoferID	CamionID	RutaID	DataComanda	Status	
1	1	1	1	1	2023-06-01	Livrat	
2	2	2	2	2	2023-06-15	In Tranzit	
3	3	3	3	3	2023-07-01	Livrat	
4	4	4	4	4	2023-07-15	Anulat	

## II. Definirea Cheilor Primare și Străine

### Raport de Execuție

Pentru a genera un raport de execuție, vom folosi SQL pentru a descrie structura tabelor, constrângerile și relațiile dintre acestea.

### 1. Descrierea Structurii Tabelor

Tabela: **Soferi**

```
DESCRIBE Soferi;
```

Rezultate:

Field	Type	Null	Key	Default	Extra
SoferID	int	NO	PRI	NULL	auto_increment
Nume	varchar(50)	YES		NULL	
Prenume	varchar(50)	YES		NULL	
DataAngajarii	date	YES		NULL	
Salariu	decimal(10,2)	YES		NULL	

```
88 • DESCRIBE Soferi;
```

```
89
```

Result Grid						
Filter Rows:						
Export:						
Wrap Cell Content:						
Field	Type	Null	Key	Default	Extra	
SoferID	int	NO	PRI	NULL	auto_increment	
Nume	varchar(50)	YES		NULL		
Prenume	varchar(50)	YES		NULL		
DataAngajarii	date	YES		NULL		
Salariu	decimal(10,2)	YES		NULL		

Tabela: **Camioane**

```
DESCRIBE Camioane;
```

Rezultate:



Field	Type	Null	Key	Default	Extra
CamionID	int	NO	PRI	NULL	auto_increment
NumarInmatriculare	varchar(20)	YES		NULL	
Marca	varchar(50)	YES		NULL	
Model	varchar(50)	YES		NULL	
AnFabricatie	int	YES		NULL	
CapacitateTonaj	decimal(5,2)	YES		NULL	

89 • DESCRIBE Camioane;

Field	Type	Null	Key	Default	Extra
CamionID	int	NO	PRI	NULL	auto_increment
NumarInmatriculare	varchar(20)	YES		NULL	
Marca	varchar(50)	YES		NULL	
Model	varchar(50)	YES		NULL	
AnFabricatie	int	YES		NULL	
CapacitateTonaj	decimal(5,2)	YES		NULL	

Tabela: **Rute**

**DESCRIBE Rute;**

Rezultate:

Field	Type	Null	Key	Default	Extra
RutaID	int	NO	PRI	NULL	auto_increment
Descriere	varchar(255)	YES		NULL	
DistantaKM	decimal(10,2)	YES		NULL	

90 • DESCRIBE Rute;

Field	Type	Null	Key	Default	Extra
RutaID	int	NO	PRI	NULL	auto_increment
Descriere	varchar(255)	YES		NULL	
DistantaKM	decimal(10,2)	YES		NULL	

Tabela: **Clienti**

**DESCRIBE Clienti;**

Rezultate:

Field	Type	Null	Key	Default	Extra
ClientID	int	NO	PRI	NULL	auto_increment
Nume	varchar(50)	YES		NULL	
Prenume	varchar(50)	YES		NULL	
Adresa	varchar(100)	YES		NULL	
Telefon	varchar(20)	YES		NULL	

91 • DESCRIBE Clienti;

Field	Type	Null	Key	Default	Extra
ClientID	int	NO	PRI	NULL	auto_increment
Nume	varchar(50)	YES		NULL	
Prenume	varchar(50)	YES		NULL	
Adresa	varchar(100)	YES		NULL	
Telefon	varchar(20)	YES		NULL	

Tabela: **Comenzi**

**DESCRIBE Comenzi;**

Rezultate:

Field	Type	Null	Key	Default	Extra
ComandaID	int	NO	PRI	NULL	auto_increment
ClientID	int	YES	MUL	NULL	
SoferID	int	YES	MUL	NULL	
CamionID	int	YES	MUL	NULL	
RutaID	int	YES	MUL	NULL	
DataComanda	date	YES		NULL	
Status	varchar(20)	YES		NULL	

92 • DESCRIBE Comenzi;

Field	Type	Null	Key	Default	Extra
ComandaID	int	NO	PRI	NULL	auto_increment
ClientID	int	YES	MUL	NULL	
SoferID	int	YES	MUL	NULL	
CamionID	int	YES	MUL	NULL	
RutaID	int	YES	MUL	NULL	
DataComanda	date	YES		NULL	
Status	varchar(20)	YES		NULL	

## 2. Relațiile dintre Tabele

- **Comenzi** este tabela centrală care leagă **Clienți**, **Soferi**, **Camioane** și **Rute**.
  - **Comenzi** are referințe la **Clienți** (ClientID), **Soferi** (SoferID), **Camioane** (CamionID) și **Rute** (RutaID) prin chei străine.
  - Relațiile sunt de tip **1** (un client poate avea mai multe comenzi, un șofer poate prelua mai multe comenzi, un camion poate fi folosit pentru mai multe comenzi, o rută poate fi folosită în mai multe comenzi).

### Relațiile 1 - explicate:

- **Tabela Clie~~n~~ți și Comenzi**: Relație: 1-n (un client (1) poate plasa mai multe comenzi (n), dar o comandă este plasată de un singur client).
  - ClientID este cheia primară în tabela **Clienți** și cheia străină în tabela **Comenzi**.
- **Tabela Soferi și Comenzi**: Relație: 1-n (un șofer (1) poate avea mai multe comenzi (n), dar o comandă este atribuită unui singur șofer)
  - SoferID este cheia primară în tabela **Soferi** și cheia străină în tabela **Comenzi**.
- **Tabela Camioane și Comenzi**: Relație: 1-n (un camion (1) poate fi folosit în mai multe comenzi (n), dar o comandă este atribuită unui singur camion)
  - CamionID este cheia primară în tabela **Camioane** și cheia străină în tabela **Comenzi**.
- **Tabela Rute și Comenzi**: Relație: 1-n (o rută (1) poate fi folosită în mai multe comenzi (n), dar o comandă este atribuită unei singure rute)
  - RutaID este cheia primară în tabela **Rute** și cheia străină în tabela **Comenzi**.

### Exemplu de utilizare a relațiilor

- Un șofer poate fi atribuit unei comenzi pentru a transporta un camion pe o anumită rută către un client specific. Astfel, în tabelul **Comenzi**, vom avea chei străine care leagă fiecare comandă de un șofer, un camion, o rută și un client. Aceste relații de tip **1** permitând urmărirea comenzilor în funcție de client, șofer, camion și rută.

Am realizat un model relațional pentru gestionarea unei companii de transport, care asigură integritatea datelor și facilitează diverse operații de management și raportare. Relațiile 1 între tabele permit urmărirea eficientă a comenzilor, șoferilor, camioanelor și rutelor asociate. Utilizarea corectă a cheilor primare și străine asigură integritatea referențială și coerența datelor în baza de date.

Acest raport de execuție din reverse engineering oferă o înțelegere clară a structurii și relațiilor dintre tabelele bazei de date, facilitând astfel întreținerea și extinderea acesteia în viitor.



### III. Gestionarea Constrângerilor și Integrității Datelor și Revenirea la Starea Inițială

Utilizare instrucțiunilor **ALTER TABLE** pentru a modifica structura tabelelor, **DROP** pentru a elimina tabele sau coloane și **TRUNCATE** pentru a goli tabelele și a metodele de revenire la starea inițială a datelor după diverse operațiuni de manipulare, inclusiv ștergerea și reinserarea datelor sau actualizarea constrângerilor.

#### 1. ALTER TABLE

Instrucțiunile **ALTER TABLE** sunt folosite pentru a modifica structura unei tabele existente, cum ar fi adăugarea sau ștergerea unei coloane, modificarea tipului de date al unei coloane sau adăugarea unei constrângeri.

Exemple de utilizare a **ALTER TABLE**:

1.1 Adăugarea unei coloane noi (**Email**) în tabela **Clients** ↔ Eliminarea coloanei **Email**

```
ALTER TABLE Clients
ADD COLUMN Email VARCHAR(100);
```



```
ALTER TABLE Clients
DROP COLUMN Email;
```

```
104 -- Adăugarea unei coloane noi (Email) în tabela Clients
105 • ALTER TABLE Clients
106 ADD COLUMN Email VARCHAR(100);
107
108 • SELECT * FROM Clients;
```

ClientID	Nume	Prenume	Adresa	Telefon	Email
1	Popa	Andrei	Strada Libertatii 10, Bucuresti	0722000000	NULL
2	Dumitrescu	Ana	Strada Independentei 20, Constanta	0722111111	NULL
3	Stan	Maria	Strada Victoriei 30, Cluj	0722222222	NULL
4	Nicolae	George	Strada Unirii 40, Timisoara	0722333333	NULL
5	NULL	NULL	NULL	NULL	NULL

```
104 -- Adăugarea unei coloane noi (Email) în tabela Clients
105 • ALTER TABLE Clients
106 ADD COLUMN Email VARCHAR(100);
107
108 • SELECT * FROM Clients;
```

```
110 -- Eliminarea coloanei Email
111 • ALTER TABLE Clients
112 DROP COLUMN Email;
```

ClientID	Nume	Prenume	Adresa	Telefon
1	Popa	Andrei	Strada Libertatii 10, Bucuresti	0722000000
2	Dumitrescu	Ana	Strada Independentei 20, Constanta	0722111111
3	Stan	Maria	Strada Victoriei 30, Cluj	0722222222
4	Nicolae	George	Strada Unirii 40, Timisoara	0722333333
5	NULL	NULL	NULL	NULL

1.2 Modificarea tipului de date pentru coloana **Telefon** în tabela **Clients** ↔ Restabilirea tipului de date original al coloanei **Telefon** la **VARCHAR(20)**

```
ALTER TABLE Clients
MODIFY COLUMN Telefon VARCHAR(15);
```



```
ALTER TABLE Clients
MODIFY COLUMN Telefon VARCHAR(20);
```

```

114 -- Modificarea tipului de date pentru coloana Telefon in tabela Clienti:
115 • ALTER TABLE Clienti
116 MODIFY COLUMN Telefon VARCHAR(15);
117
118 • SELECT * FROM Clienti;
119 • DESCRIBE Clienti;

```

Field	Type	Null	Key	Default	Extra
ClientID	int	NO	PRI	<b>HULL</b>	auto_increment
Nume	varchar(50)	YES		<b>HULL</b>	
Prenume	varchar(50)	YES		<b>HULL</b>	
Adresa	varchar(100)	YES		<b>HULL</b>	
Telefon	varchar(15)	YES		<b>HULL</b>	

```

125 -- Ștergerea unei coloane (Telefon) din tabela Clienti:
126 • ALTER TABLE Clienti
127 DROP COLUMN Telefon;
128
129 • SELECT * FROM Clienti;
130 • DESCRIBE Clienti;
131
132 -- Adăugarea coloanei Telefon din nou:
133 • ALTER TABLE Clienti
134 ADD COLUMN Telefon VARCHAR(20);

```

Field	Type	Null	Key	Default	Extra
ClientID	int	NO	PRI	<b>HULL</b>	auto_increment
Nume	varchar(50)	YES		<b>HULL</b>	
Prenume	varchar(50)	YES		<b>HULL</b>	
Adresa	varchar(100)	YES		<b>HULL</b>	
Telefon	varchar(20)	YES		<b>HULL</b>	

### 1.3 Ștergerea unei coloane (Email) din tabela Clienti ↔ Adăugarea coloanei Email din nou

```
ALTER TABLE Clienti
DROP COLUMN Email;
```



```
ALTER TABLE Clienti
ADD COLUMN Email VARCHAR(100);
```

```

125 -- Ștergerea unei coloane (Telefon) din tabela Clienti:
126 • ALTER TABLE Clienti
127 DROP COLUMN Telefon;

```

Field	Type	Null	Key	Default	Extra
ClientID	int	NO	PRI	<b>HULL</b>	auto_increment
Nume	varchar(50)	YES		<b>HULL</b>	
Prenume	varchar(50)	YES		<b>HULL</b>	
Adresa	varchar(100)	YES		<b>HULL</b>	

```

114 -- Modificarea tipului de date pentru coloana Telefon in tabela Clienti:
115 • ALTER TABLE Clienti
116 MODIFY COLUMN Telefon VARCHAR(15);
117
118 • SELECT * FROM Clienti;
119 • DESCRIBE Clienti;
120
121 -- Restabilirea tipului de date original al coloanei Telefon la VARCHAR(20):
122 • ALTER TABLE Clienti
123 MODIFY COLUMN Telefon VARCHAR(20);

```

Field	Type	Null	Key	Default	Extra
ClientID	int	NO	PRI	<b>HULL</b>	auto_increment
Nume	varchar(50)	YES		<b>HULL</b>	
Prenume	varchar(50)	YES		<b>HULL</b>	
Adresa	varchar(100)	YES		<b>HULL</b>	
Telefon	varchar(20)	YES		<b>HULL</b>	

### 1.4 Adăugarea unei constrângeri unice pentru coloana NumarInmatriculare în tabela Camioane ↔ Ștergerea constrângerii unice pentru a reveni la starea inițială

```
ALTER TABLE Camioane
ADD CONSTRAINT UNIQUE (NumarInmatriculare);
```



```
SHOW CREATE TABLE Camioane;
```

pentru a vedea toate constrângerile tabelului

```
ALTER TABLE Camioane
DROP INDEX NumarInmatriculare;
```

după identificarea numelui constrângerii, se folosește comanda **DROP INDEX**

```

136 -- Adăugarea unei constrângeri unice pentru coloana NumarInmatriculare în tabela Camioane:
137 • ALTER TABLE Camioane
138 ADD CONSTRAINT UNIQUE (NumarInmatriculare);
139
140 • DESCRIBE Camioane;

```

Field	Type	Null	Key	Default	Extra
CamionID	int	NO	PRI	<u>NULL</u>	auto_increment
NumarInmatriculare	varchar(20)	YES	UNI	<u>NULL</u>	
Marca	varchar(50)	YES		<u>NULL</u>	
Model	varchar(50)	YES		<u>NULL</u>	
AnFabricatie	int	YES		<u>NULL</u>	
CapacitateTonaj	decimal(5,2)	YES		<u>NULL</u>	

```

136 -- Adăugarea unei constrângeri unice pentru coloana NumarInmatriculare în tabela Camioane:
137 • ALTER TABLE Camioane
138 ADD CONSTRAINT UNIQUE (NumarInmatriculare);
139
140 • DESCRIBE Camioane;
141
142 -- Ștergerea constrângerii unice pentru a reveni la starea inițială:
143 • SHOW CREATE TABLE Camioane;
144
145 • ALTER TABLE Camioane
146 DROP INDEX NumarInmatriculare;

```

Field	Type	Null	Key	Default	Extra
CamionID	int	NO	PRI	<u>NULL</u>	auto_increment
NumarInmatriculare	varchar(20)	YES		<u>NULL</u>	
Marca	varchar(50)	YES		<u>NULL</u>	
Model	varchar(50)	YES		<u>NULL</u>	
AnFabricatie	int	YES		<u>NULL</u>	
CapacitateTonaj	decimal(5,2)	YES		<u>NULL</u>	

## 2. DROP TABLE

Instrucțiunea **DROP TABLE** este folosită pentru a șterge complet o tabelă din baza de date, inclusiv toate datele și structura acesteia.

### Exemple de utilizare a **DROP TABLE**:

**2.1 Ștergerea tabelului Comenzi ↔ Recrearea Tabelii** - Aceasta implică folosirea instrucțiunii **CREATE TABLE** cu aceeași definiție a structurii pe care o avea tabela **Comenzi** înainte de ștergere. Toate coloanele și relațiile (chei străine) trebuie definite la fel, readucând baza de date la starea inițială.

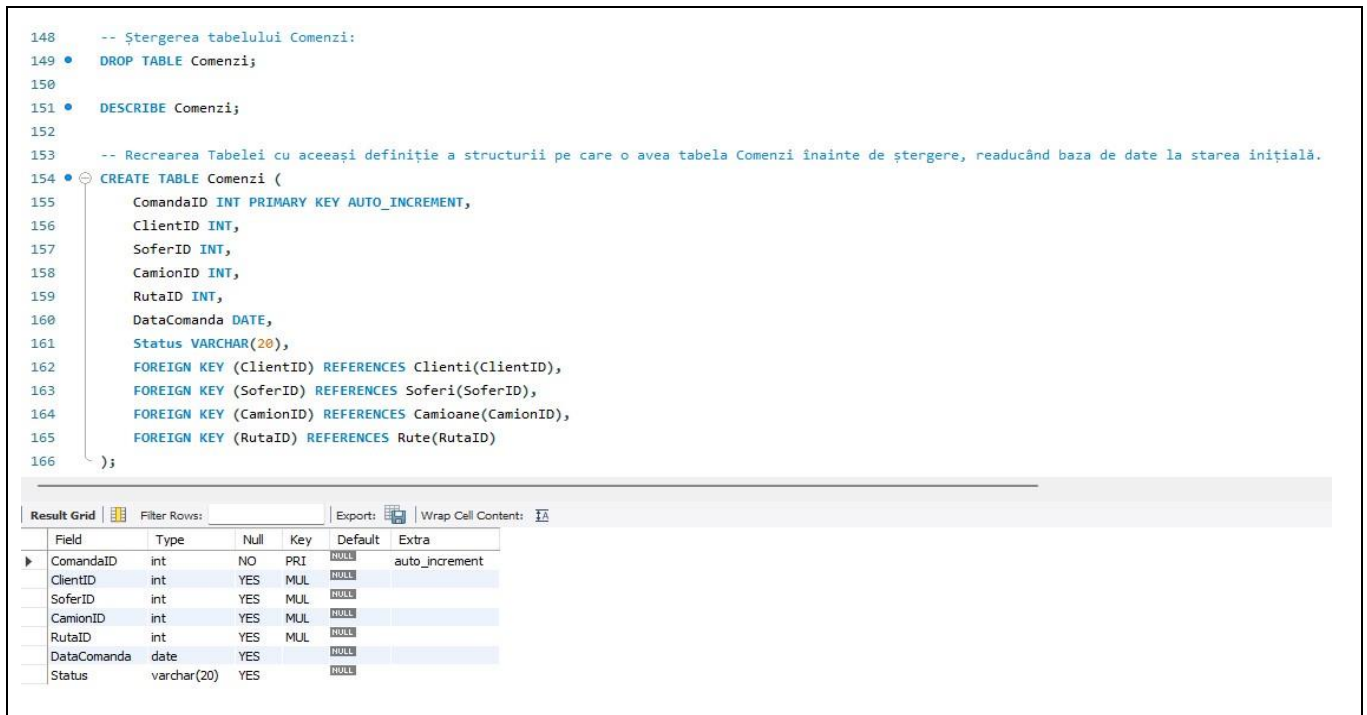
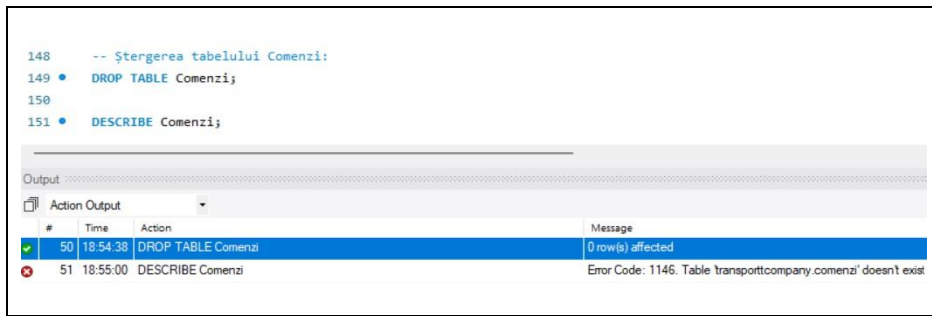
```
DROP TABLE Comenzi;
```



```

CREATE TABLE Comenzi (
    ComandaID INT PRIMARY KEY AUTO_INCREMENT,
    ClientID INT,
    SoferID INT,
    CamionID INT,
    RutaID INT,
    DataComanda DATE,
    Status VARCHAR(20),
    FOREIGN KEY (ClientID) REFERENCES Clienti(ClientID),
    FOREIGN KEY (SoferID) REFERENCES Soferi(SoferID),
    FOREIGN KEY (CamionID) REFERENCES Camioane(CamionID),
    FOREIGN KEY (RutaID) REFERENCES Rute(RutaID)
);

```



## 2.2 Ștergerea tabelului Clienti:

```
DROP TABLE Clienti;
```

## 3. TRUNCATE TABLE

Instrucțiunea **TRUNCATE TABLE** este folosită pentru a goli toate datele dintr-o tabelă, menținând structura acesteia. Este mai rapidă decât **DELETE** fără condiție pentru că nu loghează ștergerea fiecărui rând.

### Exemple de utilizare a **TRUNCATE TABLE**:

**3.1 Golirea tabelului Comenzi ↔ Reinserarea Datelor:** Pentru a reveni la starea inițială după utilizarea **TRUNCATE TABLE**, trebuie să reinserăm toate datele care au fost șterse. Aceasta presupune că avem acces la datele originale sau că le putem reintroduce manual. Prin aceste instrucțiuni, datele în tabela **Comenzi** vor fi restaurate, readucând baza de date la starea inițială înainte de folosirea **TRUNCATE TABLE**.

```
TRUNCATE TABLE Comenzi;
```



```
-- Inserarea datelor în tabela Comenzi
INSERT INTO Comenzi (ClientID, SoferID, CamionID, RutaID, DataComanda, Status) VALUES
(1, 1, 1, 1, '2023-06-01', 'Livrat'),
(2, 2, 2, 2, '2023-06-15', 'In Tranzit'),
(3, 3, 3, 3, '2023-07-01', 'Livrat'),
(4, 4, 4, 4, '2023-07-15', 'Anulat');
```

```
178 -- Golirea tabelului - se golesc toate datele din tabela Comenzi, menținând structura acesteia.
179 • TRUNCATE TABLE Comenzi;
```

ComandaID	ClientID	SoferID	CamionID	RutaID	DataComanda	Status
NULL	NULL	NULL	NULL	NULL	NULL	NULL

```
168 -- Inserarea datelor în tabela Comenzi
169 • INSERT INTO Comenzi (ClientID, SoferID, CamionID, RutaID, DataComanda, Status) VALUES
170 (1, 1, 1, 1, '2023-06-01', 'Livrat'),
171 (2, 2, 2, 2, '2023-06-15', 'In Tranzit'),
172 (3, 3, 3, 3, '2023-07-01', 'Livrat'),
173 (4, 4, 4, 4, '2023-07-15', 'Anulat');
174
175 • SELECT * FROM Comenzi;
176 • DESCRIBE Comenzi;
177
178 -- Golirea tabelului - se golesc toate datele din tabela Comenzi, menținând structura acesteia.
179 • TRUNCATE TABLE Comenzi;
```

ComandaID	ClientID	SoferID	CamionID	RutaID	DataComanda	Status
1	1	1	1	1	2023-06-01	Livrat
2	2	2	2	2	2023-06-15	In Tranzit
3	3	3	3	3	2023-07-01	Livrat
4	4	4	4	4	2023-07-15	Anulat
NULL	NULL	NULL	NULL	NULL	NULL	NULL

### 3.2 Golirea tabelului **Clienti**:

```
TRUNCATE TABLE Clienti;
```

## Explicații asupra Scenariilor și Utilizării

- **ALTER TABLE**: Este util pentru modificarea structurilor tabelelor fără a le șterge și recrea. În scenariul nostru, dacă afacerea noastră de transport decide să înregistreze emailurile clienților, putem adăuga coloana **Email** în tabela **Clienti** folosind **ALTER TABLE**. De asemenea, putem modifica tipul de date al coloanei **Telefon** pentru a asigura o lungime corectă a numărului de telefon.
- **DROP TABLE**: Este folosit pentru a șterge complet o tabelă care nu mai este necesară. De exemplu, dacă decid că tabela **Comenzi** nu mai este necesară și vrem să o eliminăm complet, vom folosi **DROP TABLE Comenzi**.

- **TRUNCATE TABLE:** Este util pentru a goli rapid o tabelă fără a șterge structura acesteia. În cazul nostru, dacă vreau să elimin toate comenzile din tabela **Comenzi** pentru a o reîncepe de la zero, voi folosi **TRUNCATE TABLE Comenzi**.

Aceste instrucțiuni adaugă flexibilitate și capacitate de întreținere în baza de date, permițându-ne să modificăm structura tabelor existente, să ștergem tabelele care nu mai sunt necesare și să golim tabelele de date pentru a le reutiliza.

## IV. Manipularea Datelor cu DML(Data Manipulation Language)

Folosirea instrucțiunilor DML (**INSERT**, **DELETE**, **UPDATE**, **SELECT**) pentru a adăuga, elimina, modifica și selecta date din tabele. Aceasta permite gestionarea în mod eficient a datelor din baza de date.

### 1. Inserarea datelor - **INSERT**

Aceste instrucțiuni de inserare adaugă date în fiecare tabel în conformitate cu relațiile definite. De exemplu, comanda va fi atribuită unui client, va folosi o anumită rută, va fi efectuată de un anumit șofer și va utiliza un anumit camion.

Inserarea datelor inițiale în tabelele **Soferi**, **Camioane**, **Rute**, **Cienti** și **Comenzi** a demonstrat cum se pot popula aceste tabele cu date reale.

```
-- Inserarea datelor în tabela Soferi
INSERT INTO Soferi (Nume, Prenume, DataAngajarii, Salariu) VALUES
('Popescu', 'Ion', '2020-01-15', 3000.00),
('Ionescu', 'Mihai', '2019-03-22', 3200.00),
('Georgescu', 'Vasile', '2018-07-10', 3100.00),
('Marinescu', 'Florin', '2021-05-30', 2900.00);

-- Inserarea datelor în tabela Camioane
INSERT INTO Camioane (NumarInmatriculare, Marca, Model, AnFabricatie, CapacitateTonaj) VAL
('B123ABC', 'Mercedes', 'Actros', 2015, 20.5),
('B234BCD', 'Volvo', 'FH16', 2018, 22.0),
('B345CDE', 'Scania', 'R450', 2017, 19.0),
('B456DEF', 'DAF', 'XF105', 2019, 21.5);
```

### 2. Selectarea datelor - **SELECT**









Aceste instrucțiuni **SELECT** recuperează datele din baza de date conform criteriilor specificate și utilizează relațiile între tabele pentru a aduna informații complete despre comenzi, inclusiv detalii despre clienți, rute, camioane și șoferi.






```
-- Selectarea tuturor datelor din tabela Soferi
SELECT * FROM Soferi;

-- Selectarea numelui și prenumele tuturor șoferilor
SELECT Nume, Prenume FROM Soferi;

-- Selectarea comenzilor livrate
SELECT * FROM Comenzi WHERE Status = 'Livrat';
```



<pre>294 -- Selectarea numelui și prenumele tuturor șoferilor 295 • SELECT Nume, Prenume FROM Soferi;</pre>	<pre>288 -- Selectarea tuturor datelor din tabela Ruta 289 • SELECT * FROM Ruta;</pre>																																		
<div><div>Result Grid</div><div> Filter Rows: <input type="text"/></div><div>Export:  Wrap Cell Content:</div></div> <table><thead><tr><th>Nume</th><th>Prenume</th></tr></thead><tbody><tr><td>▶ Popescu</td><td>Ion</td></tr><tr><td>Ionescu</td><td>Mihai</td></tr><tr><td>Georgescu</td><td>Vasile</td></tr><tr><td>Marinescu</td><td>Florin</td></tr></tbody></table>	Nume	Prenume	▶ Popescu	Ion	Ionescu	Mihai	Georgescu	Vasile	Marinescu	Florin	<div><div>Result Grid</div><div> Filter Rows: <input type="text"/></div><div>Edit:  E</div></div> <table><thead><tr><th></th><th>RutaID</th><th>Descriere</th><th>DistantaKM</th></tr></thead><tbody><tr><td>▶</td><td>1</td><td>Bucuresti - Constanta</td><td>225.50</td></tr><tr><td></td><td>2</td><td>Bucuresti - Cluj</td><td>450.00</td></tr><tr><td></td><td>3</td><td>Bucuresti - Timisoara</td><td>530.00</td></tr><tr><td></td><td>4</td><td>Bucuresti - Iasi</td><td>400.00</td></tr><tr><td>*</td><td>NULL</td><td>NULL</td><td>NULL</td></tr></tbody></table>		RutaID	Descriere	DistantaKM	▶	1	Bucuresti - Constanta	225.50		2	Bucuresti - Cluj	450.00		3	Bucuresti - Timisoara	530.00		4	Bucuresti - Iasi	400.00	*	NULL	NULL	NULL
Nume	Prenume																																		
▶ Popescu	Ion																																		
Ionescu	Mihai																																		
Georgescu	Vasile																																		
Marinescu	Florin																																		
	RutaID	Descriere	DistantaKM																																
▶	1	Bucuresti - Constanta	225.50																																
	2	Bucuresti - Cluj	450.00																																
	3	Bucuresti - Timisoara	530.00																																
	4	Bucuresti - Iasi	400.00																																
*	NULL	NULL	NULL																																

<pre>297 -- Selectarea comenzilor livrate 298 • SELECT * FROM Comenzi WHERE Status = 'Livrat';</pre>																																
<div><div>Result Grid</div><div> Filter Rows: <input type="text"/></div><div>Edit:  Export/Import:</div></div> <table><thead><tr><th></th><th>ComandaID</th><th>ClientID</th><th>SoferID</th><th>CamionID</th><th>RutaID</th><th>DataComanda</th><th>Status</th></tr></thead><tbody><tr><td>▶</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>2023-06-01</td><td>Livrat</td></tr><tr><td></td><td>3</td><td>3</td><td>3</td><td>3</td><td>3</td><td>2023-07-01</td><td>Livrat</td></tr><tr><td>*</td><td>NULL</td><td>NULL</td><td>NULL</td><td>NULL</td><td>NULL</td><td>NULL</td><td>NULL</td></tr></tbody></table>		ComandaID	ClientID	SoferID	CamionID	RutaID	DataComanda	Status	▶	1	1	1	1	1	2023-06-01	Livrat		3	3	3	3	3	2023-07-01	Livrat	*	NULL	NULL	NULL	NULL	NULL	NULL	NULL
	ComandaID	ClientID	SoferID	CamionID	RutaID	DataComanda	Status																									
▶	1	1	1	1	1	2023-06-01	Livrat																									
	3	3	3	3	3	2023-07-01	Livrat																									
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL																									

### 3. Actualizarea datelor - UPDATE

Aceste instrucțiuni actualizează datele existente în funcție de id-urile specificate, permițând modificarea capacității unui camion, adresa unui client sau data unei comenzi.

```

-- Actualizarea salariului unui șofer
UPDATE Soferi SET Salariu = 3500.00 WHERE SoferID = 1;

-- Actualizarea statusului unei comenzi
UPDATE Comenzi SET Status = 'Livrat' WHERE ComandaID = 2;

```

```
305 -- Actualizarea statusului unei comenzi
306 • UPDATE Comenzi SET Status = 'Livrat' WHERE ComandaID = 2;
```

Result Grid

Filter Rows:

Edit:

Export/Import:

	ComandaID	ClientID	SoferID	CamionID	RutaID	DataComanda	Status
▶	1	1	1	1	1	2023-06-01	Livrat
	2	2	2	2	2	2023-06-15	Livrat
	3	3	3	3	3	2023-07-01	Livrat
	4	4	4	4	4	2023-07-15	Anulat
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

```
300 -- Actualizarea salariului unui șofer
301 • UPDATE Soferi SET Salariu = 3500.00 WHERE SoferID = 1;
302
303 • SELECT * FROM Soferi;
```

Result Grid

Filter Rows:

Edit:

Export/Imp

	SoferID	Nume	Prenume	DataAngajarii	Salariu
▶	1	Popescu	Ion	2020-01-15	3500.00
	2	Ionescu	Mihai	2019-03-22	3200.00
	3	Georgescu	Vasile	2018-07-10	3100.00
	4	Marinescu	Florin	2021-05-30	2900.00
*	NULL	NULL	NULL	NULL	NULL

## 4. Ștergerea datelor - DELETE

Aceste instrucțiuni șterg înregistrările din fiecare tabel în funcție de id-urile specificate, asigurând astfel curățarea datelor vechi sau neutilizate.

```
-- Ștergerea unui client
DELETE FROM clienti WHERE ClientID = 4;

-- Ștergerea unei comenzi
DELETE FROM Comenzi WHERE ComandaID = 4;
```

```
315 -- Ștergerea unei comenzi
316 • DELETE FROM Comenzi WHERE ComandaID = 4;
317
```

ComandaID	ClientID	SoferID	CamionID	RutaID	DataComanda	Status
1	1	1	1	1	2023-06-01	Livrat
2	2	2	2	2	2023-06-15	In Tranzit
3	3	3	3	3	2023-07-01	Livrat
4	4	4	4	4	2023-07-15	Anulat

```
315 -- Ștergerea unei comenzi
316 • DELETE FROM Comenzi WHERE ComandaID = 4;
317
318 -- Inserarea comenzii nr.4
319 • INSERT INTO Comenzi (ClientID, SoferID, CamionID, RutaID, DataComanda, Status) VALUES
320 (4, 4, 4, 4, '2023-07-15', 'Anulat');
```

ComandaID	ClientID	SoferID	CamionID	RutaID	DataComanda	Status
1	1	1	1	1	2023-06-01	Livrat
2	2	2	2	2	2023-06-15	In Tranzit
3	3	3	3	3	2023-07-01	Livrat
4	4	4	4	4	2023-07-15	Anulat

Am realizat operații de inserare, actualizare și ștergere a datelor pentru a gestiona informațiile din tabele.

## V. Instrucțiuni DQL(Data Query Language)

### Interogări și Manipularea Datelor

Aceste interogări și operații acoperă diverse scenarii și demonstrează cum putem manipula și interoga datele pentru a obține informații relevante și valoroase.

**Select All** - Selectarea tuturor coloanelor dintr-o tabelă.

```
-- Selectarea tuturor datelor din toate tabelele
SELECT * FROM Soferi;
SELECT * FROM Camioane;
SELECT * FROM Rute;
SELECT * FROM Clienti;
SELECT * FROM Comenzi;
```

```
181 -- Selectarea tuturor datelor din tabele
182 • SELECT * FROM Soferi;
183 • SELECT * FROM Camioane;
```

CamionID	NumarInmatriculare	Marca	Model	AnFabricatie	CapacitateTonaj
1	B123ABC	Mercedes	Actros	2015	20.50
2	B234BCD	Volvo	FH16	2018	22.00
3	B345CDE	Scania	R450	2017	19.00
4	B456DEF	DAF	XF105	2019	21.50

```
181 -- Selectarea tuturor datelor din tabele
182 • SELECT * FROM Soferi;
```

SoferID	Nume	Prenume	DataAngajarii	Salariu
1	Popescu	Ion	2020-01-15	3000.00
2	Ionescu	Mihai	2019-03-22	3200.00
3	Georgescu	Vasile	2018-07-10	3100.00
4	Marinescu	Florin	2021-05-30	2900.00

```
181 -- Selectarea tuturor datelor din tabele
182 • SELECT * FROM Soferi;
183 • SELECT * FROM Camioane;
184 • SELECT * FROM Rute;
185 • SELECT * FROM Clienti;
```

ClientID	Nume	Prenume	Adresa	Telefon
1	Popa	Andrei	Strada Libertatii 10, Bucuresti	0722000000
2	Dumitrescu	Ana	Strada Independentei 20, Constanta	0722111111
3	Stan	Maria	Strada Victoriei 30, Cluj	0722222222
4	Nicolae	George	Strada Unirii 40, Timisoara	0722333333

## Select Câteva Coloane - Selectarea unor coloane specifice dintr-o tabelă.

```
-- Selectarea numelui și prenumele tuturor șoferilor
SELECT Nume, Prenume FROM Soferi;

-- Selectarea numelui și adresei tuturor clienților
SELECT Nume, Adresa FROM Clienti;

-- Selectarea numărului de înmatriculare și marca camioanelor
SELECT NumarInmatriculare, Marca FROM Camioane;
```

192

-- Selectarea numelui și adresei tuturor clienților

193

• SELECT Nume, Adresa FROM Clienti;

Result Grid

Filter Rows:

Export:

Wrap Cell Contents:

	Nume	Adresa
▶	Popa	Strada Libertati 10, Bucuresti
	Dumitrescu	Strada Independentei 20, Constanta
	Stan	Strada Victoriei 30, Cluj
	Nicolae	Strada Unirii 40, Timisoara

192

-- Selectarea numelui și prenumele tuturor șoferilor

193

• SELECT Nume, Prenume FROM Soferi;

194

Result Grid

Filter Rows:

Export:

Wrap Cell Contents:

	Nume	Prenume
▶	Popescu	Ion
	Ionescu	Mihai
	Georgescu	Vasile
	Marinescu	Florin

195

-- Selectarea numărului de înmatriculare și marca camioanelor

196

• SELECT NumarInmatriculare, Marca FROM Camioane;

Result Grid

Filter Rows:

Export:

Wrap Cell Contents:

	NumarInmatriculare	Marca
▶	B123ABC	Mercedes
	B234BCD	Volvo
	B345CDE	Scania
	B456DEF	DAF

## Filtrare cu WHERE - Filtrarea rezultatelor pe baza unei condiții.

```
-- Selectarea șoferilor cu salariul mai mare de 3000
SELECT * FROM Soferi WHERE Salariu > 3000;

-- Selectarea comenzilor cu statusul 'Livrat'
SELECT * FROM Comenzi WHERE Status = 'Livrat';
```

198

-- Selectarea șoferilor cu salariul mai mare de 3000

199

• SELECT \* FROM Soferi WHERE Salariu > 3000;

Result Grid

Filter Rows:

Edit:

Export/Imp

	SoferID	Nume	Prenume	DataAngajarii	Salariu
▶	2	Ionescu	Mihai	2019-03-22	3200.00
	3	Georgescu	Vasile	2018-07-10	3100.00
•	NULL	NULL	NULL	NULL	NULL

201

-- Selectarea comenzilor cu statusul 'Livrat'

202

• SELECT \* FROM Comenzi WHERE Status = 'Livrat';

Result Grid

Filter Rows:

Edit:

Export/Import:

	ComandaID	ClientID	SoferID	CamionID	RutaID	DataComanda	Status
▶	1	1	1	1	1	2023-06-01	Livrat
	3	3	3	3	3	2023-07-01	Livrat
•	NULL	NULL	NULL	NULL	NULL	NULL	NULL

## Filtrare cu LIKE - Utilizarea LIKE căuta modele în date text.

```
-- Selectarea clienților cu numele care începe cu 'P'
SELECT * FROM Clienti WHERE Nume LIKE 'P%';

-- Selectarea camioanelor cu marca ce conține 'Volvo'
SELECT * FROM Camioane WHERE Marca LIKE '%Volvo%';
```

210 -- Selectarea camioanelor cu marca ce conține 'Volvo'

211 • SELECT \* FROM Camioane WHERE Marca LIKE '%Volvo%';

Result Grid

Filter Rows:

Edit:

Export/Import

CamionID	NumarInmatriculare	Marca	Model	AnFabricatie	CapacitateTonaj
2	B234BCD	Volvo	FH16	2018	22.00
NULL	NULL	NULL	NULL	NULL	NULL

204 -- Selectarea clienților cu numele care începe cu 'P'

205 • SELECT \* FROM Clienti WHERE Nume LIKE 'P%';

Result Grid

Filter Rows:

Edit:

Export/Import

ClientID	Nume	Prenume	Adresa	Telefon
1	Popa	Andrei	Strada Libertatii 10, Bucuresti	0722000000
NULL	NULL	NULL	NULL	NULL

207 -- Selectarea clienților cu adresa ca 'Strada'

208 • SELECT \* FROM Clienti WHERE Adresa LIKE 'Strada%';

Result Grid

Filter Rows:

Edit:

Export/Import

ClientID	Nume	Prenume	Adresa	Telefon
1	Popa	Andrei	Strada Libertatii 10, Bucuresti	0722000000
2	Dumitrescu	Ana	Strada Independentei 20, Constanta	0722111111
3	Stan	Maria	Strada Victoriei 30, Cluj	0722222222
4	Nicolae	George	Strada Unirii 40, Timisoara	0722333333
NULL	NULL	NULL	NULL	NULL

## Filtrare cu AND și OR - Combinarea condițiilor de filtrare.

```
-- Selectarea șoferilor angajați după 2020 și cu salariul mai mare de 3000
SELECT * FROM Soferi WHERE DataAngajarii > '2020-01-01' AND Salariu > 3000;

-- Selectarea comenzilor livrate sau în tranzit
SELECT * FROM Comenzi WHERE Status = 'Livrat' OR Status = 'In Tranzit';
```

216 -- Selectarea comenzilor livrate sau în tranzit 217 • SELECT * FROM Comenzi WHERE Status = 'Livrat' OR Status = 'In Tranzit';	213 -- Selectarea șoferilor angajați după 2020 și cu salariul mai mare de 3000 214 • SELECT * FROM Soferi WHERE DataAngajarii > '2020-01-01' AND Salariu > 3000;
--	---

ComandaID	ClientID	SoferID	CamionID	RutaID	DataComanda	Status
1	1	1	1	1	2023-06-01	Livrat
2	2	2	2	2	2023-06-15	In Tranzit
3	3	3	3	3	2023-07-01	Livrat
4	4	4	4	4	2023-07-15	Anulat

SoferID	Nume	Prenume	DataAngajarii	Salariu
1	Popescu	Ion	2020-01-15	3000.00
2	Marinescu	Florin	2021-05-30	2900.00
3	Popescu	Ion	2020-01-15	3000.00
4	Marinescu	Florin	2021-05-30	2900.00

## Filtrare cu NOT - Excluderea rezultatelor care îndeplinesc o condiție.

```
-- Selectarea șoferilor care nu au salariul mai mare de 3000
SELECT * FROM Soferi WHERE NOT Salariu > 3000;

-- Selectarea comenzilor care nu sunt livrate
SELECT * FROM Comenzi WHERE NOT Status = 'Livrat';
```

219 -- Selectarea șoferilor care nu au salariul mai mare de 3000 220 • SELECT * FROM Soferi WHERE NOT Salariu > 3000;	225 -- Selectarea comenzilor care nu sunt livrate 226 • SELECT * FROM Comenzi WHERE NOT Status = 'Livrat';	222 -- Selectarea camioanelor care nu sunt marca Volvo 223 • SELECT * FROM Camioane WHERE NOT Marca = 'Volvo';
--	---	---

SoferID	Nume	Prenume	DataAngajarii	Salariu
1	Popescu	Ion	2020-01-15	3000.00
2	Marinescu	Florin	2021-05-30	2900.00
3	Popescu	Ion	2020-01-15	3000.00
4	Marinescu	Florin	2021-05-30	2900.00

ComandaID	ClientID	SoferID	CamionID	RutaID	DataComanda	Status
2	2	2	2	2	2023-06-15	In Tranzit
4	4	4	4	4	2023-07-15	Anulat
3	3	3	3	3	2023-07-01	Livrat
1	1	1	1	1	2023-06-01	Livrat

CamionID	NumarInmatriculare	Marca	Model	AnFabricatie	CapacitateTona
1	B123ABC	Mercedes	Actros	2015	20.50
3	B345CDE	Scania	R450	2017	19.00
4	B456DEF	DAF	XF105	2019	21.50
2	B234GHI	Volvo	FM260	2016	25.00

## Funcții Agregate - Utilizarea funcțiilor pentru a calcula valori cumulate și GROUP BY - Gruparea rezultatelor pentru a aplica funcții agregate.

```
-- Numărul total de comenzi
SELECT COUNT(*) AS NumarComenzi FROM Comenzi;

-- Salariul mediu al șoferilor
SELECT AVG(Salariu) AS SalariuMediu FROM Soferi;

-- Numărul de comenzi per client
SELECT ClientID, COUNT(*) AS NumarComenzi FROM Comenzi GROUP BY ClientID;
```

231 -- Salariul mediu al șoferilor 232 • SELECT AVG(Salariu) AS SalariuMediu FROM Soferi;	228 -- Numărul total de comenzi 229 • SELECT COUNT(*) AS NumarComenzi FROM Comenzi;
--	--

SalariuMediu
3050.000000

NumarComenzi
4



```

234 -- Numărul de comenzi per client
235 • SELECT ClientID, COUNT(*) AS NumarComenzi FROM Comenzi GROUP BY ClientID;

```

ClientID	NumarComenzi
1	1
2	1
3	1
4	1

**Filtrare pe Funcții Agregate cu HAVING** - Filtrarea rezultatelor grupate pe baza unei funcții agregate.

```

-- Numărul de comenzi per șofer cu cel puțin 2 comenzi
SELECT SoferID, COUNT(*) AS NumarComenzi
FROM Comenzi
GROUP BY SoferID
HAVING COUNT(*) >= 2;

-- Salariul mediu al șoferilor angajați după 2020
SELECT DataAngajarii, AVG(Salariu) AS SalariuMediu
FROM Soferi
GROUP BY DataAngajarii
HAVING DataAngajarii > '2020-01-01';

```

```

243 -- Salariul mediu al șoferilor angajați după 2020
244 • SELECT DataAngajarii, AVG(Salariu) AS SalariuMediu
245 FROM Soferi
246 GROUP BY DataAngajarii
247 HAVING DataAngajarii > '2020-01-01';

```

DataAngajarii	SalariuMediu
2020-01-15	3000.000000
2021-05-30	2900.000000

## Join-uri

**INNER JOIN** - Alăturarea tabelor pentru a selecta datele comune.

```

-- Selectarea detaliilor comenzilor, șoferilor și clienților pentru comenzi livrate
SELECT C.ComandaID, Cl.Nume AS NumeClient, S.Nume AS NumeSofer, S.Prenume AS PrenumeSofer
FROM Comenzi C
INNER JOIN Cienti Cl ON C.ClientID = Cl.ClientID
INNER JOIN Soferi S ON C.SoferID = S.SoferID
WHERE C.Status = 'Livrat';

```

```

251 -- Selectarea detaliilor comenzilor, șoferilor și clienților pentru comenzi livrate
252 • SELECT C.ComandaID, Cl.Nume AS NumeClient, S.Nume AS NumeSofer, S.Prenume AS PrenumeSofer, C.Status
253 FROM Comenzi C
254 INNER JOIN Cienti Cl ON C.ClientID = Cl.ClientID
255 INNER JOIN Soferi S ON C.SoferID = S.SoferID
256 WHERE C.Status = 'Livrat';

```

ComandaID	NumeClient	NumeSofer	PrenumeSofer	Status
1	Popa	Popescu	Ion	Livrat
3	Stan	Georgescu	Vasile	Livrat

**LEFT JOIN** - Alăturarea tabelor pentru a selecta toate datele din tabela stângă și datele comune din tabela dreaptă.

```

-- Selectarea tuturor șoferilor și comenzilor asociate, dacă există
SELECT S.Nume, S.Prenume, C.ComandaID, C.Status
FROM Soferi S
LEFT JOIN Comenzi C ON S.SoferID = C.SoferID;

```

```

258 -- Selectarea tuturor șoferilor și comenzilor asociate, dacă există
259 • SELECT S.Nume, S.Prenume, C.ComandaID, C.Status
260 FROM Soferi S
261 LEFT JOIN Comenzi C ON S.SoferID = C.SoferID;

```

Nume	Prenume	ComandaID	Status
Popescu	Ion	1	Livrat
Ionescu	Mihai	2	In Tranzit
Georgescu	Vasile	3	Livrat
Marinescu	Florin	4	Anulat

**RIGHT JOIN** - Alăturarea tabelor pentru a selecta toate datele din tabela dreaptă și datele comune din tabela stângă.

```

-- Selectarea tuturor comenzilor și șoferilor asociați, dacă există
SELECT C.ComandaID, C.Status, S.Nume, S.Prenume
FROM Comenzi C
RIGHT JOIN Soferi S ON C.SoferID = S.SoferID;

```

```

263 -- Selectarea tuturor comenzilor și șoferilor asociați, dacă există
264 • SELECT C.ComandaID, C.Status, S.Nume, S.Prenume
265 FROM Comenzi C
266 RIGHT JOIN Soferi S ON C.SoferID = S.SoferID;

```

ComandaID	Status	Nume	Prenume
1	Livrat	Popescu	Ion
2	In Tranzit	Ionescu	Mihai
3	Livrat	Georgescu	Vasile
4	Anulat	Marinescu	Florin

**CROSS JOIN** - Crearea produsului cartezian între două tabele.

```

-- Generarea tuturor combinațiilor posibile de șoferi și camioane
SELECT S.Nume AS NumeSofer, S.Prenume AS PrenumeSofer, C.Marca AS MarcaCamion, C.Model AS ModelCamion
FROM Soferi S
CROSS JOIN Camioane C;

```

```

268 -- Generarea tuturor combinațiilor posibile de șoferi și camioane
269 • SELECT S.Nume AS NumeSofer, S.Prenume AS PrenumeSofer, C.Marca AS MarcaCamion, C.Model AS ModelCamion
270 FROM Soferi S
271 CROSS JOIN Camioane C;

```

NumeSofer	PrenumeSofer	MarcaCamion	ModelCamion
Marinescu	Florin	Mercedes	Actros
Georgescu	Vasile	Mercedes	Actros
Ionescu	Mihai	Mercedes	Actros
Popescu	Ion	Mercedes	Actros
Marinescu	Florin	Volvo	FH16
Georgescu	Vasile	Volvo	FH16
Ionescu	Mihai	Volvo	FH16
Popescu	Ion	Volvo	FH16
Marinescu	Florin	Scania	R450
Georgescu	Vasile	Scania	R450
Ionescu	Mihai	Scania	R450
Popescu	Ion	Scania	R450
Marinescu	Florin	DAF	XF105
Georgescu	Vasile	DAF	XF105
Ionescu	Mihai	DAF	XF105
Popescu	Ion	DAF	XF105



## Limite și Ordine

**LIMIT și ORDER BY** - Limitarea numărului de rezultate și ordonarea rezultatelor.

```
-- Selectarea primilor 2 șoferi în ordinea descrescătoare a salariului
SELECT * FROM Soferi ORDER BY Salariu DESC LIMIT 2;

-- Selectarea primelor 3 comenzi în ordinea crescătoare a datei comenzii
SELECT * FROM Comenzi ORDER BY DataComanda ASC LIMIT 3;
```

```
274 -- Selectarea primilor 2 șoferi în ordinea descrescătoare a salariului
275 • SELECT * FROM Soferi ORDER BY Salariu DESC LIMIT 2;
```

SoferID	Nume	Prenume	DataAngajarii	Salariu
2	Ionescu	Mihai	2019-03-22	3200.00
3	Georgescu	Vasile	2018-07-10	3100.00
NULL	NULL	NULL	NULL	NULL

```
277 -- Selectarea primelor 3 comenzi în ordinea crescătoare a datei comenzii
278 • SELECT * FROM Comenzi ORDER BY DataComanda ASC LIMIT 3;
```

ComandaID	ClientID	SoferID	CamionID	RutaID	DataComanda	Status
1	1	1	1	1	2023-06-01	Livrat
2	2	2	2	2	2023-06-15	In Tranzit
3	3	3	3	3	2023-07-01	Livrat
NULL	NULL	NULL	NULL	NULL	NULL	NULL

### \*SUBQUERY

```
287 -- Selectarea camioanelor care au fost utilizate pentru comenzi în tranzit
288 • SELECT *
289 FROM Camioane
290 WHERE CamionID IN (SELECT CamionID FROM Comenzi WHERE Status = 'In Tranzit');
```

CamionID	NumarInmatriculare	Marca	Model	AnFabricatie	CapacitateTonaj
2	B234BCD	Volvo	FH16	2018	22.00
NULL	NULL	NULL	NULL	NULL	NULL

```
282 -- Selectarea șoferilor care au efectuat cel puțin o comanda livrată
283 • SELECT *
284 FROM Soferi
285 WHERE SoferID IN (SELECT SoferID FROM Comenzi WHERE Status = 'Livrat');
```

SoferID	Nume	Prenume	DataAngajarii	Salariu
1	Popescu	Ion	2020-01-15	3000.00
3	Georgescu	Vasile	2018-07-10	3100.00
NULL	NULL	NULL	NULL	NULL

Prin utilizarea acestor funcții și tehnici SQL, proiectul de gestionare a transporturilor devine eficient și capabil să manipuleze și să analizeze datele complex într-o manieră organizată și integrată. Fiecare instrucțiune joacă un rol esențial în menținerea integrității și a funcționalității bazei de date.

## Concluzie Finală a Proiectului de Gestionare a Transporturilor

În acest proiect, am creat o bază de date pentru o companie de transport, incluzând tabele pentru șoferi, camioane, rute, clienți și comenzi. Am demonstrat utilizarea instrucțiunilor DDL, DML și DQL pentru a crea, manipula și interoga datele din aceste tabele.

### Învățăături Cheie

- **Structurarea și Designul Bazei de Date:** Importanța unei structuri bine definite pentru a facilita gestionarea și interogarea datelor.
- **Integritatea Datelor:** Utilizarea cheilor primare și străine pentru a menține integritatea referențială.
- **Manipularea și Interogarea Datelor:** Cum se folosesc eficient instrucțiunile SQL pentru a adăuga, elimina, modifica și selecta date.
- **Complexitatea Joinurilor:** Importanța și aplicabilitatea diverselor tipuri de joinuri pentru a combina date din tabele multiple.
- **Gestionarea Constrângerilor:** Cum se modifică structura bazei de date și se gestionează constrângerile pentru a asigura integritatea datelor.

### Concluzie Finală

Prin acest proiect, am demonstrat crearea unei baze de date complexe și interogarea acesteia folosind diverse tipuri de instrucțiuni SQL. Am construit tabele și relații între acestea, utilizând chei primare și chei străine pentru a asigura integritatea referențială a datelor. Am utilizat instrucțiuni de manipulare și interogare a datelor pentru a extrage informații relevante și pentru a gestiona datele într-o manieră eficientă. Am demonstrat utilizarea funcțiilor agregate, a filtrărilor complexe și a diferitelor tipuri de join-uri pentru a răspunde la diverse cerințe de business.