

Jegyzőkönyv

Adatbázisrendszerek I.

Féléves feladat: Autó kölcsönző

Készítette: **Buha Milán**
Neptunkód: **IY5AM2**
Gyakorlat időpontja: **Kedd 10-12**
Gyakorlatvezető: **Dr. Bednarik László**

Miskolc, 2022

Tartalomjegyzék

1. A feladat leírása	2
2. Adatbázis ER modellje	3
3. Adatbázis konvertálása relációs modellre	4
4. Adatbázis relációs modellje	5
5. Adatbázis relációs sémája	6
6. Adattáblák létrehozása	7
7. Adattáblák feltöltése	14
8. Lekérdezések	24
8.1. Lekérdezés I.	24
8.2. Lekérdezés II.	25
8.3. Lekérdezés III.	26
8.4. Lekérdezés IV.	27
8.5. Lekérdezés V.	28
8.6. Lekérdezés VI.	29
8.7. Lekérdezés VII.	30
8.8. Lekérdezés VIII.	31
8.9. Lekérdezés IX.	32
8.10. Lekérdezés X.	33

1. A feladat leírása

Adatbázisom egy autó kölcsönző/fuvarozó cég hálózatát mutatja be, amiben helyet kapnak a cég alkalmazottjai, a bérlők, fuvarozók, autók és a tranzakciók. Az egyedek neveit angolul adtam meg, mivel angol a programozás "alap" nyelve.

Összesen 5 egyedet hoztam létre, amelyek a következők:

- **Transaction**
- **Cars**
- **Driver**
- **Admin**
- **Customer**

A **Transaction** egyedből indul ki minden. Itt tárolódnak a legfőbb tulajdonságok, pl: a tranzakció azonosítója, neve, ideje és típusa. Az elsődleges kulcsa a `transaction_id`, ami a tranzakció azonosítója.

A **Cars** és **Transaction** között 1:N kapcsolat van, mivel egy autóhoz több tranzakció is tartozhat, viszont egy adott tranzakcióhoz csak egyetlen autó. Ezen kapcsolat 1:N kapcsolat neve *Selecting*.

A **Cars** egyednek 5 db tulajdonsága van: `car_id`, `car_model`, `rent_price`, `car_number`, `car_status`. Az elsődleges kulcsa a `car_id`, amivel azonosíthatjuk az autót. Ezen felül a `car_model` az autó típusát, a `rent_price` az autó óránkénti bérleti árát, a `car_number` a rendszámát, illetve a `car_status` az autó elérhetőségét adja meg.

A **Driver** és **Cars** között 1:1 kapcsolat van, mert egy sofőrnek csak egyetlen autója lehet, a kapcsolat neve: *Choosing_driver*.

A **Driver** egyednek már több tulajdonsága is van, a legfontosabb a `driver_id`, ami az elsődleges kulcs. Megtalálható a vezetéki és keresztnév (ami az ER modellben egy többágú tulajdonságként szerepel), telefonszáma, egy "address" nevezetű többágú tulajdonság, ami tárolja az irányító számot, várost és az utcát. Valamint egy tulajdonság, ami a nemét tárolja.

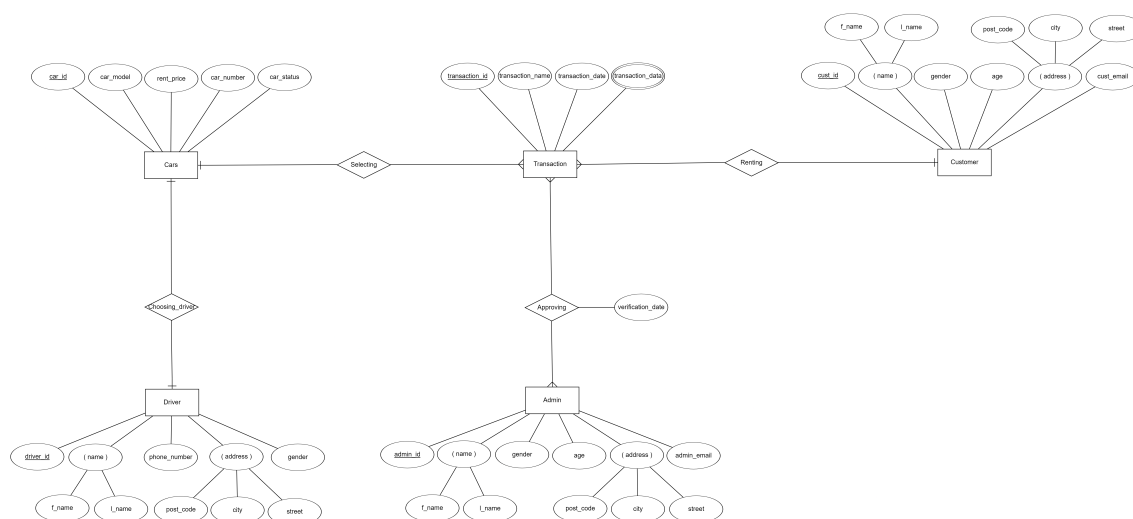
Az **Admin** és **Transaction** egyedek között egy *Approving* nevű N:M kapcsolat áll fent, aminek van egy tulajdonsága `verification_date`. Fontos kiemelni, hogy a kapcsolat külön kapcsolótáblát fog kapni a relációs modellben. Az **Admin** egyednek egy hasonló tulajdonságai vannak, mint a **Driver** egyednek. Van azonosítója, ami az elsődleges tulajdonsága, vezetéki és keresztnév, neme, kora, egy "address" többágú tulajdonsága és email címe.

Végül az utolsó egyed következik, a **Customer**. A **Transaction** és **Customer** egyedek között 1:N kapcsolat áll fent, amelynek a neve *Renting*.

A **Customer** egyednek az **Admin** egyedhez képes ugyan azok a tulajdonságai: `cust_id` nevű tulajdonság, ami az elsődleges kulcsa, vezetéki és keresztnév, neme, kora, egy "address" többágú tulajdonság, illetve egy email társul.

2. Adatbázis ER modellje

A legelső entitás, amit létrehoztam fontossága miatt a **Transaction** volt, mivel a többi entitás rajta keresztül kapcsolódik egymáshoz, illetve neki van a legtöbb kapcsolata. Így kifele haladva könnyen elrendezhető a többi entitás körülötte.



3. Adatbázis konvertálása relációs modellre

Jobbról balra haladva, előbb létrehozzuk a **Customer** táblát. A táblában nem található foreign key. Itt a customer_id az elsődleges kulcs, ami egyben AUTO_INCREMENT, tehát automatikus növekedik. A többágú tulajdonságot ketté bontjuk és így kapjuk a first_name és last_name tulajdonságokat, végezetül ezek VARCHAR(30) típusú column-ök. A gender tulajdonság CHAR(1) típusú, mivel az angol "Male" és "Female" szavak kezdőbetűit kell megadni értéknek. ('M';'F') Age tulajdonságnak INT(2) van beállítva. Végül maradtak a post_code, city, street és email columnok, amelyek mint VARCHAR típusúak 30-40 karakterre korlátozva.

Következőnek jön a **Transaction** tábla, itt a többértékű tulajdonsághoz egy külön táblát kell rendelnünk (Transaction_transaction_data), ahol a foreign key lesz a transaction_id. A transaction_data pedig a többértékű tulajdonságunk, ami két féle értéket vehet fel. CASH vagy CARD, vagyis a fizetési módot tárolja. Az eredeti táblában a transaction_id INT az elsődleges kulcs, ami szintén AUTO_INCREMENT. A transaction_name column a tranzakció nevét tárolja: transaction: + transaction_id VARCHAR(30) formátumban. Végül van a transaction_date, ami DATE formátumú, ráadásul NOT NULL értéket vesz fel, mint ahogy minden column mindegyik táblában. Két foreign key található benne, amik összekötni a **Transaction** táblát a **Customer**-rel és **Cars** táblákkal, amelyek az alábbiak: customer_id és car_id.

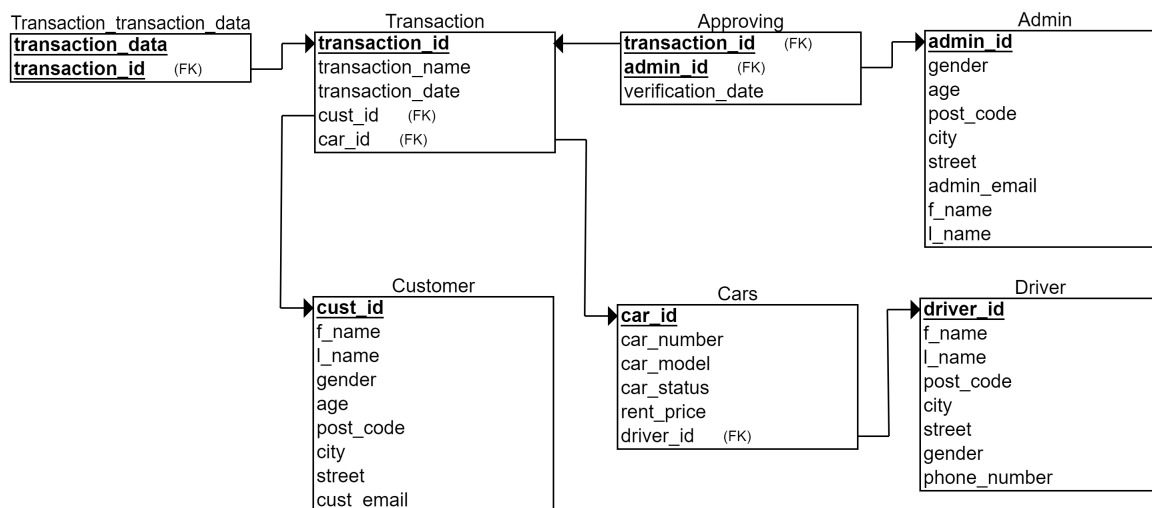
Ezen az ágon már csak egyetlen tábla maradt, ami a **Driver**. Itt nem található foreign key. A driver_id az elsődleges kulcs, illetve AUTO_INCREMENT. Található benne first_name, last_name, post_code, city, street, phone_number és gender columnok, amelyek VARCHAR(30) típust vesznek fel, kivéve a gender (CHAR(1)) ÉS phone_number (VARCHAR 12).

Az N:M kapcsolat relációs táblájával folytatjuk, aminek két idegenkulcsa van, amik: transaction_id és Admin_id. Illetve tartalmaz egy verification_date nevű column-ot, ami DATE típusú és NOT NULL. Így tudjuk összekötni az **Admin** és **Transaction** táblákat.

Végül jönnek a cégben dolgozók táblája az **Admin**. Fontos megemlítenem, hogy MySQL Workbench használja az Admin szót, ezért AdminA-nak neveztem el. A táblában nem található foreign key. Az elsődleges kulcs itt Admin_id, illetve AUTO_INCREMENT. A szokásos first_name VARCHAR(30), last_name VARCHAR(30), post_code VARCHAR(30), city VARCHAR(30), street VARCHAR(30), age INT(2), gender CHAR(1) és email VARCHAR(40) column-ök találhatók benne.

4. Adatbázis relációs modellje

Jobbról balra haladva hozom létre azokat a táblákat, amelyekre mutat idegen kulcs, utána pedig csak azokat, amik elsődleges kulcsokat tartalmaznak.



5. Adatbázis relációs sémája

Customer [customer_id, first_name, last_name, gender, age, post_code, city, street, email]

Driver [driver_id, first_name, last_name, post_code, city, street, gender, phone_number]

Cars [car_id, car_number, model, car_status, rent_price, driver_id]

Trans [transaction_id, transaction_name, transaction_date, car_id, customer_id]

Trans_data [transaction_data, transaction_id]

AdminA [AdminA_id, first_name, last_name, gender, age, post_code, city, street, email]

Approving [verification_date, transaction_id, AdminA_id]

6. Adattáblák létrehozása

Az adattáblák létrehozásánál ügyelni kell a helyes sorrendre. Én előbb azokat a táblákat hoztam létre, amire mutat idegenkulcs, utána pedig azokat, amik csak elsődleges kulcsokat tartalmaznak.

```

1 • DROP DATABASE IF EXISTS RS;
2 • CREATE DATABASE RS;
3
4 • DROP TABLE IF EXISTS RS.Customer;
5 • CREATE TABLE RS.Customer(
6     customer_id INT PRIMARY KEY AUTO_INCREMENT,
7     first_name VARCHAR(30) NOT NULL,
8     last_name VARCHAR(30) NOT NULL,
9     gender CHAR(1) NOT NULL,
10    age INT(2) NOT NULL,
11    post_code VARCHAR(30) NOT NULL,
12    city VARCHAR(30) NOT NULL,
13    street VARCHAR(30) NOT NULL,
14    email VARCHAR(40) NOT NULL
15 );
16 • DESCRIBE RS.Customer;

```


Result Grid | Filter Rows: | Export: | Wrap Cell C

	Field	Type	Null	Key	Default	Extra
▶	customer_id	int	NO	PRI	NULL	auto_increment
	first_name	varchar(30)	NO		NULL	
	last_name	varchar(30)	NO		NULL	
	gender	char(1)	NO		NULL	
	age	int	NO		NULL	
	post_code	varchar(30)	NO		NULL	
	city	varchar(30)	NO		NULL	
	street	varchar(30)	NO		NULL	
	email	varchar(40)	NO		NULL	


```

18 • DROP TABLE IF EXISTS RS.Driver;
19 • CREATE TABLE RS.Driver(
20     driver_id INT PRIMARY KEY AUTO_INCREMENT,
21     first_name VARCHAR(30) NOT NULL,
22     last_name VARCHAR(30) NOT NULL,
23     post_code VARCHAR(30) NOT NULL,
24     city VARCHAR(30) NOT NULL,
25     street VARCHAR(30) NOT NULL,
26     gender CHAR(1) NOT NULL,
27     phone_number VARCHAR(12) NOT NULL
28 );
29 • DESCRIBE RS.Driver;

```

<						
Result Grid						
Filter Rows: <input type="text"/>						
Export:  Wrap Cell Cont						
	Field	Type	Null	Key	Default	Extra
▶	driver_id	int	NO	PRI	NULL	auto_increment
	first_name	varchar(30)	NO		NULL	
	last_name	varchar(30)	NO		NULL	
	post_code	varchar(30)	NO		NULL	
	city	varchar(30)	NO		NULL	
	street	varchar(30)	NO		NULL	
	gender	char(1)	NO		NULL	
	phone_number	varchar(12)	NO		NULL	

```

31 • DROP TABLE IF EXISTS RS.Cars;
32 • CREATE TABLE RS.Cars(
33     car_id INT PRIMARY KEY AUTO_INCREMENT,
34     car_number VARCHAR(7) NOT NULL,
35     model VARCHAR(30) NOT NULL,
36     car_status VARCHAR(1) NOT NULL,
37     rent_price INT NOT NULL,
38     driver_id INT,
39     FOREIGN KEY(driver_id) REFERENCES RS.Driver(driver_id) ON DELETE CASCADE
40 );
41 • DESCRIBE RS.Cars;



```

Result Grid Filter Rows: Export: Wrap Cell Content: iA						
	Field	Type	Null	Key	Default	Extra
►	car_id	int	NO	PRI	NULL	auto_increment
	car_number	varchar(7)	NO		NULL	
	model	varchar(30)	NO		NULL	
	car_status	varchar(1)	NO		NULL	
	rent_price	int	NO		NULL	
	driver_id	int	YES	MUL	NULL	

```

43 • DROP TABLE IF EXISTS RS.Trans;
44 • CREATE TABLE RS.Trans(
45     transaction_id INT PRIMARY KEY AUTO_INCREMENT,
46     transaction_name VARCHAR(30) NOT NULL,
47     transaction_date DATE NOT NULL,
48     car_id INT,
49     customer_id INT,
50     FOREIGN KEY(customer_id) REFERENCES RS.Customer(customer_id) ON DELETE CASCADE,
51     FOREIGN KEY(car_id) REFERENCES RS.Cars(car_id) ON DELETE CASCADE
52 );
53 • DESCRIBE RS.Trans;

```

Result Grid						
Filter Rows: <input type="text"/>						
Export:  Wrap Cell Content: 						
	Field	Type	Null	Key	Default	Extra
▶	transaction_id	int	NO	PRI	NULL	auto_increment
	transaction_name	varchar(30)	NO		NULL	
	transaction_date	date	NO		NULL	
	car_id	int	YES	MUL	NULL	
	customer_id	int	YES	MUL	NULL	











```
55 • DROP TABLE IF EXISTS RS.Trans_data;  
56 • CREATE TABLE RS.Trans_data(  
57     transaction_data VARCHAR(4),  
58     transaction_id INT,  
59     FOREIGN KEY(transaction_id) REFERENCES RS.Trans(transaction_id) ON DELETE CASCADE  
60 );  
61 • DESCRIBE RS.Trans_data;
```

Result Grid						
		Filter Rows:			Export:	Wrap Cell Content:
Field	Type	Null	Key	Default	Extra	
transaction_data	varchar(4)	YES		NULL		
transaction_id	int	YES	MUL	NULL		

```

63 • DROP TABLE IF EXISTS RS.AdminA;
64 • CREATE TABLE RS.AdminA(
65     AdminA_id INT PRIMARY KEY AUTO_INCREMENT,
66     first_name VARCHAR(30) NOT NULL,
67     last_name VARCHAR(30) NOT NULL,
68     gender CHAR(1) NOT NULL,
69     age INT(2) NOT NULL,
70     post_code VARCHAR(30) NOT NULL,
71     city VARCHAR(30) NOT NULL,
72     street VARCHAR(30) NOT NULL,
73     email VARCHAR(40) NOT NULL
74 );
75 • DESCRIBE RS.AdminA;

```

Result Grid						
		Filter Rows:	Export: 			
	Field	Type	Null	Key	Default	Extra
▶	AdminA_id	int	NO	PRI		auto_increment
	first_name	varchar(30)	NO			
	last_name	varchar(30)	NO			
	gender	char(1)	NO			
	age	int	NO			
	post_code	varchar(30)	NO			
	city	varchar(30)	NO			
	street	varchar(30)	NO			
	email	varchar(40)	NO			

```
77 • DROP TABLE IF EXISTS RS.Approving;
78 • CREATE TABLE RS.Approving(
79     verification_date DATE NOT NULL,
80     transaction_id INT,
81     AdminA_id INT,
82     FOREIGN KEY(transaction_id) REFERENCES RS.Trans(transaction_id) ON DELETE CASCADE,
83     FOREIGN KEY(AdminA_id) REFERENCES RS.AdminA(AdminA_id) ON DELETE CASCADE
84 );
85 • DESCRIBE RS.Approving;
```

Field	Type	Null	Key	Default	Extra
verification_date	date	NO		NULL	
transaction_id	int	YES	MUL	NULL	
AdminA_id	int	YES	MUL	NULL	

7. Adattáblák feltöltése

```

1 • INSERT INTO RS.Customer VALUES(1, 'Buha', 'Milán', 'M', 21, 3534, 'Miskolc', 'Árpád', 'buha360@gmail.com');
2 • INSERT INTO RS.Customer VALUES(2, 'Kovács', 'Lajos', 'M', 25, 3534, 'Miskolc', 'Kis Ernő', 'kovacs-lajos25@gmail.com');
3 • INSERT INTO RS.Customer VALUES(3, 'Kis', 'Norbert', 'M', 32, 4031, 'Debrecen', 'Szabadság', 'kisinorbert32@gmail.com');
4 • INSERT INTO RS.Customer VALUES(4, 'Nagy', 'Lídia', 'F', 26, 4044, 'Debrecen', 'Kuruc', 'nagyliidia26@gmail.com');
5 • INSERT INTO RS.Customer VALUES(5, 'Kovács', 'Dorina', 'F', 35, 3535, 'Miskoc', 'Testvériség', 'kovacs-dorina35@gmail.com');
6 • INSERT INTO RS.Customer VALUES(6, 'Hunyadi', 'Mátyás', 'M', 57, 3528, 'Miskolc', 'Árpád', 'hunyadi-matyas57@gmail.com');
7 • INSERT INTO RS.Customer VALUES(7, 'Kalecsár', 'József', 'M', 45, 3538, 'Miskolc', 'Béke', 'kalecsar-jozsef45@gmail.com');
8 • INSERT INTO RS.Customer VALUES(8, 'Pánkucsi', 'Barbara', 'F', 24, 3902, 'Szerencs', 'Rákóczi', 'pankucsi-barbara24@gmail.com');
9 • INSERT INTO RS.Customer VALUES(9, 'Esze', 'László', 'M', 27, 3903, 'Szerencs', 'Szent István', 'esz-laszlo27@gmail.com');
10 • INSERT INTO RS.Customer VALUES(10, 'Kopasz', 'Dominika', 'F', 24, 3300, 'Eger', 'Deák Ferenc', 'kopasz-dominika24@gmail.com');
11 • INSERT INTO RS.Customer VALUES(11, 'Szakács', 'Gergő', 'M', 31, 3903, 'Szerencs', 'Almagyár', 'szakacs-gergo31@gmail.com');
12 • INSERT INTO RS.Customer VALUES(12, 'Hajdú', 'István', 'M', 43, 3302, 'Eger', 'Zöldfa', 'hajdu-istvan43@gmail.com');
13 • INSERT INTO RS.Customer VALUES(13, 'Kovács', 'István', 'M', 40, 3300, 'Eger', 'Alkotmány', 'kovacs-istvan40@gmail.com');
14 • INSERT INTO RS.Customer VALUES(14, 'Nagy', 'Dénes', 'M', 50, 3536, 'Miskolc', 'Andor', 'nagydenes50@gmail.com');
15 • INSERT INTO RS.Customer VALUES(15, 'Kis', 'Heni', 'F', 31, 4040, 'Debrecen', 'Báthory', 'kisheni31@gmail.com');
16 • INSERT INTO RS.Customer VALUES(16, 'Szörös', 'Márton', 'M', 36, 4037, 'Debrecen', 'Boldog', 'szorosmarton36@gmail.com');
17 • INSERT INTO RS.Customer VALUES(17, 'Éles', 'Barbara', 'F', 25, 3534, 'Miskolc', 'Csavar', 'halmivanessza25@gmail.com');
18 • SELECT * FROM RS.Customer;

```

Result Grid								
Filter Rows: <input type="text"/>								
Edit: Export/Import: Wrap Cell Content:								
customer_id	first_name	last_name	gender	age	post_code	city	street	email
1	Buha	Milán	M	21	3534	Miskolc	Árpád	buha360@gmail.com
2	Kovács	Lajos	M	25	3534	Miskolc	Kis Ernő	kovacs-lajos25@gmail.com
3	Kis	Norbert	M	32	4031	Debrecen	Szabadság	kisinorbert32@gmail.com
4	Nagy	Lídia	F	26	4044	Debrecen	Kuruc	nagyliidia26@gmail.com
5	Kovács	Dorina	F	35	3535	Miskoc	Testvériség	kovacs-dorina35@gmail.com
6	Hunyadi	Mátyás	M	57	3528	Miskolc	Árpád	hunyadi-matyas57@gmail.com
7	Kalecsár	József	M	45	3538	Miskolc	Béke	kalecsar-jozsef45@gmail.com
8	Pánkucsi	Barbara	F	24	3902	Szerencs	Rákóczi	pankucsi-barbara24@gmail.com
9	Esze	László	M	27	3903	Szerencs	Szent István	esz-laszlo27@gmail.com
10	Kopasz	Dominika	F	24	3300	Eger	Deák Ferenc	kopasz-dominika24@gmail.com
11	Szakács	Gergő	M	31	3903	Szerencs	Almagyár	szakacs-gergo31@gmail.com
12	Hajdú	István	M	43	3302	Eger	Zöldfa	hajdu-istvan43@gmail.com
13	Kovács	István	M	40	3300	Eger	Alkotmány	kovacs-istvan40@gmail.com
14	Nagy	Dénes	M	50	3536	Miskolc	Andor	nagydenes50@gmail.com
15	Kis	Heni	F	31	4040	Debrecen	Báthory	kisheni31@gmail.com
16	Szörös	Márton	M	36	4037	Debrecen	Boldog	szorosmarton36@gmail.com
17	Éles	Barbara	F	25	3534	Miskolc	Csavar	halmivanessza25@gmail.com

```

20 • INSERT INTO RS.Driver VALUES(1, 'Michael', 'Schumacher', 3534, 'Miskolc', 'Árpád', 'M', "06501311078");
21 • INSERT INTO RS.Driver VALUES(2, 'Sebastian', 'Vettel', 4042, 'Debrecen', 'Kívánság', 'M', "06305643298");
22 • INSERT INTO RS.Driver VALUES(3, 'Vass', 'Gábor', 3530, 'Miskolc', 'Csermőkei', 'M', "06708235546");
23 • INSERT INTO RS.Driver VALUES(4, 'Soós', 'Sára', 3903, 'Szerencs', 'Szarkahegy', 'F', "06208573349");
24 • INSERT INTO RS.Driver VALUES(5, 'Budai', 'Nikoletta', 4040, 'Debrecen', 'Dicsősség', 'F', "0620353449");
25 • INSERT INTO RS.Driver VALUES(6, 'Lakatos', 'Bence', 3537, 'Miskolc', 'Nagy István', 'M', "06503458546");
26 • INSERT INTO RS.Driver VALUES(7, 'Gál', 'Vilmos', 3300, 'Eger', 'Boldogság', 'M', "06707453359");
27 • INSERT INTO RS.Driver VALUES(8, 'Fülöp', 'Dalma', 3302, 'Eger', 'Tatár', 'F', "06508763995");
28 • SELECT * FROM RS.Driver;






```

Result Grid								
Filter Rows: <input type="text"/>								
Edit: Export/Import: Wrap Cell Content:								
	driver_id	first_name	last_name	post_code	city	street	gender	phone_number
▶	1	Michael	Schumacher	3534	Miskolc	Árpád	M	06501311078
	2	Sebastian	Vettel	4042	Debrecen	Kívánság	M	06305643298
	3	Vass	Gábor	3530	Miskolc	Csermőkei	M	06708235546
	4	Soós	Sára	3903	Szerencs	Szarkahegy	F	06208573349
	5	Budai	Nikoletta	4040	Debrecen	Dicsősség	F	0620353449
	6	Lakatos	Bence	3537	Miskolc	Nagy István	M	06503458546
	7	Gál	Vilmos	3300	Eger	Boldogság	M	06707453359
	8	Fülöp	Dalma	3302	Eger	Tatár	F	06508763995


```

30 • INSERT INTO RS.Cars VALUES(1, 'RTG-663', 'Ford Focus', 'A', 40, 3);
31 • INSERT INTO RS.Cars VALUES(2, 'EHB-925', 'Renault Twingo', 'A', 30, 5);
32 • INSERT INTO RS.Cars VALUES(3, 'PQA-443', 'Chevrolet Camaro', 'A', 110, 8);
33 • INSERT INTO RS.Cars VALUES(4, 'PRA-429', 'LaFerrari', 'A', 250, 1);
34 • INSERT INTO RS.Cars VALUES(5, 'PUC-802', 'Chevrolet Corvette', 'A', 140, 6);
35 • INSERT INTO RS.Cars VALUES(6, 'ERA-084', 'Opel Corsa', 'A', 25, 4);
36 • INSERT INTO RS.Cars VALUES(7, 'RVA-382', 'Seat Leon', 'U', 20, 7);
37 • INSERT INTO RS.Cars VALUES(8, 'EJA-469', 'Aston Martin Vulcan', 'A', 180, 2);
38 • SELECT * FROM RS.Cars;

```

Result Grid						
Filter Rows: <input type="text"/>						
Edit:   						
Export/Import:  						
Wrap Cell Cont						
	car_id	car_number	model	car_status	rent_price	driver_id
▶	1	RTG-663	Ford Focus	A	40	3
	2	EHB-925	Renault Twingo	A	30	5
	3	PQA-443	Chevrolet Camaro	A	110	8
	4	PRA-429	LaFerrari	A	250	1
	5	PUC-802	Chevrolet Corvette	A	140	6
	6	ERA-084	Opel Corsa	A	25	4
	7	RVA-382	Seat Leon	U	20	7
	8	EJA-469	Aston Martin Vulcan	A	180	2








```
40 • INSERT INTO RS.Trans VALUES(1, 'Transaction:1', '2022-11-20', 3, 1);
41 • INSERT INTO RS.Trans VALUES(2, 'Transaction:2', '2022-11-20', 1, 2);
42 • INSERT INTO RS.Trans VALUES(3, 'Transaction:3', '2022-11-21', 4, 3);
43 • INSERT INTO RS.Trans VALUES(4, 'Transaction:4', '2022-11-24', 2, 4);
44 • INSERT INTO RS.Trans VALUES(5, 'Transaction:5', '2022-11-24', 8, 17);
45 • INSERT INTO RS.Trans VALUES(6, 'Transaction:6', '2022-11-25', 8, 5);
46 • INSERT INTO RS.Trans VALUES(7, 'Transaction:7', '2022-11-25', 2, 6);
47 • INSERT INTO RS.Trans VALUES(8, 'Transaction:8', '2022-11-25', 5, 7);
48 • INSERT INTO RS.Trans VALUES(9, 'Transaction:9', '2022-11-25', 8, 1);
49 • INSERT INTO RS.Trans VALUES(10, 'Transaction:10', '2022-11-26', 3, 8);
50 • INSERT INTO RS.Trans VALUES(11, 'Transaction:11', '2022-11-26', 4, 10);
51 • INSERT INTO RS.Trans VALUES(12, 'Transaction:12', '2022-11-27', 1, 4);
52 • INSERT INTO RS.Trans VALUES(13, 'Transaction:13', '2022-11-27', 8, 15);
53 • INSERT INTO RS.Trans VALUES(14, 'Transaction:14', '2022-11-27', 4, 13);
54 • INSERT INTO RS.Trans VALUES(15, 'Transaction:15', '2022-11-28', 5, 17);
55 • INSERT INTO RS.Trans VALUES(16, 'Transaction:16', '2022-11-29', 4, 16);
56 • INSERT INTO RS.Trans VALUES(17, 'Transaction:17', '2022-11-29', 2, 14);
57 • INSERT INTO RS.Trans VALUES(18, 'Transaction:18', '2022-11-29', 2, 9);
58 • INSERT INTO RS.Trans VALUES(19, 'Transaction:19', '2022-11-30', 2, 10);
59 • INSERT INTO RS.Trans VALUES(20, 'Transaction:20', '2022-11-30', 2, 11);
60 • INSERT INTO RS.Trans VALUES(21, 'Transaction:21', '2022-11-30', 2, 12);
61 • INSERT INTO RS.Trans VALUES(22, 'Transaction:22', '2022-11-30', 2, 14);
62 • SELECT * FROM RS.Trans;
```

	transaction_id	transaction_name	transaction_dat	car_id	customer_id
▶	1	Transaction:1	2022-11-20	3	1
	2	Transaction:2	2022-11-20	1	2
	3	Transaction:3	2022-11-21	4	3
	4	Transaction:4	2022-11-24	2	4
	5	Transaction:4	2022-11-24	8	17
	6	Transaction:6	2022-11-25	8	5
	7	Transaction:7	2022-11-25	2	6
	8	Transaction:8	2022-11-25	5	7
	9	Transaction:9	2022-11-25	8	1
	10	Transaction:10	2022-11-26	3	8
	11	Transaction:11	2022-11-26	4	10
	12	Transaction:12	2022-11-27	1	4
	13	Transaction:13	2022-11-27	8	15
	14	Transaction:14	2022-11-27	4	13
	15	Transaction:15	2022-11-28	5	17
	16	Transaction:16	2022-11-29	4	16
	17	Transaction:17	2022-11-29	2	14
	18	Transaction:18	2022-11-29	2	9
	19	Transaction:19	2022-11-30	2	10
	20	Transaction:20	2022-11-30	2	11
	21	Transaction:21	2022-11-30	2	12
	22	Transaction:22	2022-11-30	2	14

```
64 • INSERT INTO RS.Trans_data VALUES('Cash', 1);
65 • INSERT INTO RS.Trans_data VALUES('Cash', 2);
66 • INSERT INTO RS.Trans_data VALUES('Card', 3);
67 • INSERT INTO RS.Trans_data VALUES('Card', 4);
68 • INSERT INTO RS.Trans_data VALUES('Cash', 5);
69 • INSERT INTO RS.Trans_data VALUES('Cash', 6);
70 • INSERT INTO RS.Trans_data VALUES('Card', 7);
71 • INSERT INTO RS.Trans_data VALUES('Cash', 8);
72 • INSERT INTO RS.Trans_data VALUES('Card', 9);
73 • INSERT INTO RS.Trans_data VALUES('Card', 10);
74 • INSERT INTO RS.Trans_data VALUES('Cash', 11);
75 • INSERT INTO RS.Trans_data VALUES('Card', 12);
76 • INSERT INTO RS.Trans_data VALUES('Cash', 13);
77 • INSERT INTO RS.Trans_data VALUES('Card', 14);
78 • INSERT INTO RS.Trans_data VALUES('Card', 15);
79 • INSERT INTO RS.Trans_data VALUES('Cash', 16);
80 • INSERT INTO RS.Trans_data VALUES('Cash', 17);
81 • INSERT INTO RS.Trans_data VALUES('Cash', 18);
82 • INSERT INTO RS.Trans_data VALUES('Card', 19);
83 • INSERT INTO RS.Trans_data VALUES('Card', 20);
84 • INSERT INTO RS.Trans_data VALUES('Cash', 21);
85 • INSERT INTO RS.Trans_data VALUES('Cash', 22);
86 • SELECT * FROM RS.Trans_data;
```

	transaction_data	transaction_id
►	Cash	1
	Cash	2
	Card	3
	Card	4
	Cash	5
	Cash	6
	Card	7
	Cash	8
	Card	9
	Card	10
	Cash	11
	Card	12
	Cash	13
	Card	14
	Card	15
	Cash	16
	Cash	17
	Cash	18
	Card	19
	Card	20
	Cash	21
	Cash	22

```
88 • INSERT INTO RS.AdminA VALUES(1, 'Király', 'Norbert', 'M', 29, 3535, 'Miskolc', 'Csipkevirág', 'kiralynorber29@gmail.com');
89 • INSERT INTO RS.AdminA VALUES(2, 'Nagy', 'Barnabás', 'M', 37, 3900, 'Szerencs', 'Erdész', 'nagybarnabas37@gmail.com');
90 • INSERT INTO RS.AdminA VALUES(3, 'Szeles', 'Tekla', 'F', 24, 4040, 'Debrecen', 'Havas', 'szelestekla24@gmail.com');
91 • SELECT * FROM RS.AdminA;
```

Result Grid									
Filter Rows: <input type="text"/>									
Edit:    									
Export/Import:  									
Wrap Cell Content: 									
AdminA_id	first_name	last_name	gender	age	post_code	city	street	email	
1	Király	Norbert	M	29	3535	Miskolc	Csipkevirág	kiralynorber29@gmail.com	
2	Nagy	Barnabás	M	37	3900	Szerencs	Erdész	nagybarnabas37@gmail.com	
3	Szeles	Tekla	F	24	4040	Debrecen	Havas	szelestekla24@gmail.com	

```
93 • INSERT INTO RS.Approving VALUES('2022-11-20', 1, 1);
94 • INSERT INTO RS.Approving VALUES('2022-11-20', 2, 2);
95 • INSERT INTO RS.Approving VALUES('2022-11-21', 3, 3);
96 • INSERT INTO RS.Approving VALUES('2022-11-24', 4, 3);
97 • INSERT INTO RS.Approving VALUES('2022-11-24', 5, 2);
98 • INSERT INTO RS.Approving VALUES('2022-11-25', 6, 3);
99 • INSERT INTO RS.Approving VALUES('2022-11-25', 7, 1);
100 • INSERT INTO RS.Approving VALUES('2022-11-25', 8, 3);
101 • INSERT INTO RS.Approving VALUES('2022-11-25', 9, 2);
102 • INSERT INTO RS.Approving VALUES('2022-11-26', 10, 2);
103 • INSERT INTO RS.Approving VALUES('2022-11-26', 11, 3);
104 • INSERT INTO RS.Approving VALUES('2022-11-27', 12, 1);
105 • INSERT INTO RS.Approving VALUES('2022-11-27', 13, 3);
106 • INSERT INTO RS.Approving VALUES('2022-11-27', 14, 1);
107 • INSERT INTO RS.Approving VALUES('2022-11-28', 15, 2);
108 • INSERT INTO RS.Approving VALUES('2022-11-29', 16, 1);
109 • INSERT INTO RS.Approving VALUES('2022-11-29', 17, 2);
110 • INSERT INTO RS.Approving VALUES('2022-11-29', 18, 3);
111 • INSERT INTO RS.Approving VALUES('2022-11-30', 19, 2);
112 • INSERT INTO RS.Approving VALUES('2022-11-30', 20, 1);
113 • INSERT INTO RS.Approving VALUES('2022-11-30', 21, 3);
114 • INSERT INTO RS.Approving VALUES('2022-11-30', 22, 2);
115 • SELECT * FROM RS.Approving;
```

	verification_date	transaction_id	AdminA_id
►	2022-11-20	1	1
	2022-11-20	2	2
	2022-11-21	3	3
	2022-11-24	4	3
	2022-11-24	5	2
	2022-11-25	6	3
	2022-11-25	7	1
	2022-11-25	8	3
	2022-11-25	9	2
	2022-11-26	10	2
	2022-11-26	11	3
	2022-11-27	12	1
	2022-11-27	13	3
	2022-11-27	14	1
	2022-11-28	15	2
	2022-11-29	16	1
	2022-11-29	17	2
	2022-11-29	18	3
	2022-11-30	19	2
	2022-11-30	20	1
	2022-11-30	21	3
	2022-11-30	22	2

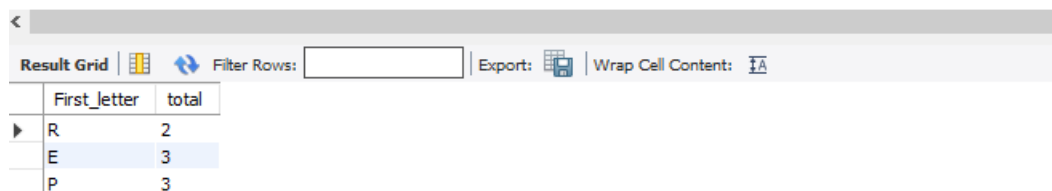
8. Lekérdezések

8.1. Lekérdezés I.

Kérdezzük le, hogy hány olyan rendszám van, amelyeknek a kezdőbetűi megegyeznek ?

$$\pi_{car_number \bowtie first_letter, COUNT(*) \bowtie total}$$
$$\gamma_{COUNT(*) cars}$$

```
1 • SELECT LEFT(RS.Cars.car_number, 1) AS First_letter, COUNT(*) AS total FROM RS.Cars
2   GROUP BY First_letter;
3
```



The screenshot shows a database interface with a 'Result Grid' tab. The grid displays the results of the SQL query, showing three rows of data. The first row is highlighted with a blue background. The columns are 'First_letter' and 'total'.

	First_letter	total
▶	R	2
	E	3
	P	3

8.2. Lekérdezés II.

Kérdezzük le, hogy az adatbázisban összesen mennyi nő szerepel ?

$\Gamma_{COUNT(*)}$

```
1 • SELECT
2   (SELECT COUNT(gender) FROM RS.Customer WHERE gender = 'F') +
3   (SELECT COUNT(gender) FROM RS.Driver WHERE gender = 'F') +
4   (SELECT COUNT(gender) FROM RS.AdminA WHERE gender = 'F')
5   AS Sum_of_females;
```



The screenshot shows a database query result grid. The grid has one column labeled 'Sum_of_females' and one row with the value '10'. Above the grid, there is a toolbar with options for 'Result Grid', 'Filter Rows', 'Export', and 'Wrap Cell Content'.

Sum_of_females
10

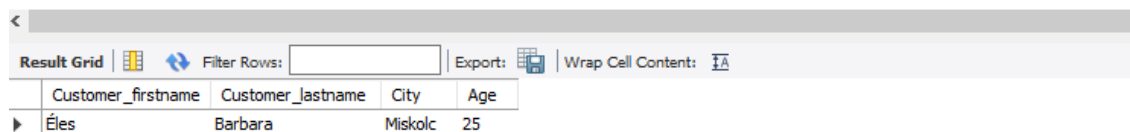
8.3. Lekérdezés III.

Kérdezzük le, hogy van-e olyan személy az adatbázisban, aki nő, miskolci és fiatalabb, mint 30 ?

$\pi_{first_name \parallel customer_firstname, last_name \parallel customer_lastname, city \parallel city, age \parallel age}$

$\sigma_{gender = 'F' \text{ AND } city = 'Miskolc' \text{ AND } age \leq 30}$

```
1 • SELECT RS.Customer.first_name AS Customer_firstname, RS.Customer.last_name AS Customer_lastname,  
2     RS.Customer.city AS City, RS.Customer.age AS Age FROM RS.Customer  
3  
4     WHERE gender = 'F' AND city = 'Miskolc' AND age <= 30;
```



The screenshot shows a database query result grid. The grid has four columns: Customer_firstname, Customer_lastname, City, and Age. The first row of data shows the name Éles, the last name Barbara, the city Miskolc, and the age 25. The grid is titled 'Result Grid' and includes a 'Filter Rows' field, an 'Export' button, and a 'Wrap Cell Content' checkbox.

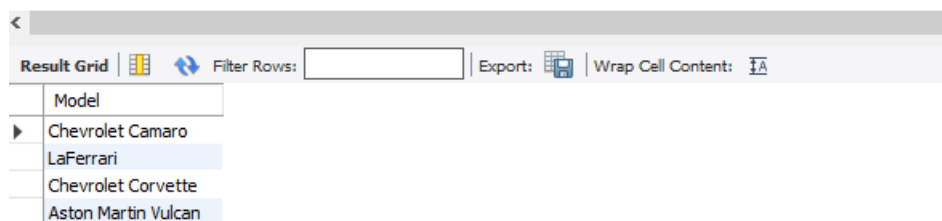
	Customer_firstname	Customer_lastname	City	Age
▶	Éles	Barbara	Miskolc	25

8.4. Lekérdezés IV.

Kérdezzük le mennyi olyan autó van, amelyek drágábban a többi autó átlag áránál!

$$\pi_{model \rightarrow Model}$$
$$\sigma_{(rent_price > (\pi \text{ AVG}(rent_price)))}$$

```
1 • SELECT RS.Cars.model AS Model FROM RS.Cars
2   GROUP BY rent_price
3   HAVING RS.Cars.rent_price > (SELECT AVG(RS.Cars.rent_price) from RS.Cars);
```



The screenshot shows a database query result grid. The grid has a header row with the column 'Model'. Below the header, there are four rows of data: 'Chevrolet Camaro', 'LaFerrari', 'Chevrolet Corvette', and 'Aston Martin Vulcan'. The 'LaFerrari' row is highlighted. Above the grid, there is a toolbar with icons for 'Result Grid', 'Filter Rows', 'Export', and 'Wrap Cell Content'.

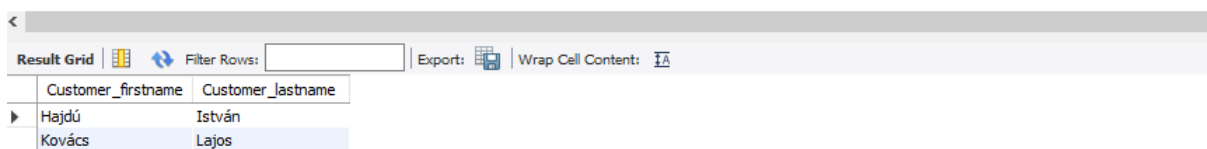
Model
Chevrolet Camaro
LaFerrari
Chevrolet Corvette
Aston Martin Vulcan

8.5. Lekérdezés V.

Kérdezzük le, hogy ki/kik azok a férfi Customerek, akik az 1.es ID-jű autót kérték ?

$$\pi_{first_name \bowtie customer_firstname, last_name \bowtie customer_lastname}$$
$$\sigma_{gender='M' \wedge car_id=1(customertransaction_id=customer_idtrans)}$$

```
1 • SELECT RS.Customer.first_name AS Customer_firstname, RS.Customer.last_name AS Customer_lastname FROM RS.Customer
2 JOIN RS.Trans ON Trans.transaction_id = Customer.customer_id
3 WHERE gender = 'M' AND car_id = 1
4 ORDER BY Customer.first_name;
```



Customer_firstname	Customer_lastname
Hajdú	István
Kovács	Lajos

8.6. Lekérdezés VI.

Kérdezzük le, hogy ki/kik azok a Customerek, akik az alábbi (2022-11-25) napon tranzakciót hajtottak végre és a fizetési módjukat ?

$$\pi_{first_name \beta customer_firstname, last_name \beta customer_lastname, transaction_id \beta transaction_id, transaction_data \beta pay_method}$$

$$\sigma_{transaction_date = '2022-11-25'}(transcustomer_id = transaction_id customer transaction_id = transaction_id trans_data)$$

```

1 • SELECT RS.Customer.first_name AS Customer_firstname, RS.Customer.last_name AS Customer_lastname,
2     RS.Trans.transaction_id AS Transaction_id, RS.Trans_data.transaction_data AS Pay_method FROM RS.Trans
3
4     JOIN RS.Customer ON Customer.customer_id = Trans.transaction_id
5     JOIN RS.Trans_data ON Trans_data.transaction_id = Trans.transaction_id
6     WHERE transaction_date = '2022-11-25'
7     ORDER BY Customer.first_name;

```

Result Grid				
Filter Rows:		Export:		
Customer_firstname	Customer_lastname	Transaction_id	Pay_method	
Esze	László	9	Card	
Hunyadi	Mátyás	6	Cash	
Kalecsár	József	7	Card	
Pánkucsi	Barbara	8	Cash	

8.7. Lekérdezés VII.

Kérdezzük le a rendszerben lévő sofőrök neveit, hozzájuk tartozó telefonszámaikat, autó modelleket, elérhető-e az autó és melyik városban tartózkodnak ?

$$\pi_{first_name \parallel driver_firstname, last_name \parallel driver_lastname, city \parallel city, phone_number \parallel phone_number, model \parallel car} \sigma_{car_status = 'A'}(driver \parallel driver_id = driver_id \parallel cars)$$

```

1 • SELECT RS.Driver.first_name AS Driver_firstname, RS.Driver.last_name AS Driver_lastname, RS.Driver.city as City,
2     RS.Driver.phone_number AS Phone_number, RS.Cars.model AS Car FROM RS.Driver
3
4 JOIN RS.Cars ON RS.Cars.driver_id = RS.Driver.driver_id
5 WHERE Cars.car_status = 'A'
6 ORDER BY Driver.first_name;
```

	Driver_firstname	Driver_lastname	City	Phone_number	Car
▶	Budai	Nikoletta	Debrecen	0620353449	Renault Twingo
	Fülöp	Dalma	Eger	06508763995	Chevrolet Camaro
	Lakatos	Bence	Miskolc	06503458546	Chevrolet Corvette
	Michael	Schumacher	Miskolc	06501311078	LaFerrari
	Sebastian	Vettel	Debrecen	06305643298	Aston Martin Vulcan
	Soós	Sára	Szerencs	06208573349	Opel Corsa
	Vass	Gábor	Miskolc	06708235546	Ford Focus

8.8. Lekérdezés VIII.

Kérdezzük le, hogy mennyien fizetek CASH-el!

$$\pi_{COUNT(transaction_data) \text{ as } payed_with_card}$$

$$\gamma_{COUNT(transaction_data)}$$

$$\sigma_{transaction_data = 'CASH'}(trans_data \text{ transaction_id } = transaction_idtrans)$$

```
1 • SELECT COUNT(RS.Trans_data.transaction_data) AS Payed_with_card FROM RS.Trans_data
2 JOIN RS.Trans ON Trans.transaction_id = Trans_data.transaction_id
3 WHERE Trans_data.transaction_data = 'CASH';
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	Payed_with_card			
▶	12			

8.9. Lekérdezés IX.

Kérdezzük le, hogy egy adott időintervallumon belül az adminok mely tranzakciókat hitelesítették!

$\pi_{first_name \parallel admin_firstname, last_name \parallel admin_lastname, transaction_id \parallel transaction_id}$

$\sigma_{"2022-11-25" \leq verification_date \text{ AND } verification_date \leq "2022-12-28"}(admin \parallel admin_id = admin_id \parallel approving \parallel transaction_id = tr$

```

1 • SELECT RS.AdminA.first_name AS Admin_firstname, RS.AdminA.last_name AS Admin_lastname,
2     RS.Trans.transaction_id AS Transaction_id FROM RS.AdminA
3
4     JOIN RS.Approving ON Approving.AdminA_id = AdminA.AdminA_id
5     JOIN RS.Trans ON Trans.transaction_id = Approving.transaction_id
6 WHERE Approving.verification_date BETWEEN '2022-11-25' AND '2022-12-28'
7 ORDER BY Trans.transaction_id;

```

Result Grid			
Filter Rows: <input type="text"/>			
Export: <input type="text"/>			
Wrap Cell Content: <input type="text"/>			
	Admin_firstname	Admin_lastname	Transaction_id
▶	Szeles	Tekla	6
	Király	Norbert	7
	Szeles	Tekla	8
	Nagy	Barnabás	9
	Nagy	Barnabás	10
	Szeles	Tekla	11
	Király	Norbert	12
	Szeles	Tekla	13
	Király	Norbert	14
	Nagy	Barnabás	15
	Király	Norbert	16
	Nagy	Barnabás	17
	Szeles	Tekla	18
	Nagy	Barnabás	19
	Király	Norbert	20
	Szeles	Tekla	21
	Nagy	Barnabás	22

8.10. Lekérdezés X.

Kérdezzük le, hogy a női Customerek mely Driverekkel utaztak egy adott időintervallumban!

$\pi_{first_name\beta customer_firstname, last_name\beta customer_lastname, first_name\beta driver_firstname, last_name\beta driver_lastname}$

$\sigma_{"2022-11-22" \leq transaction_date \wedge transaction_date \leq "2022-12-26" \wedge gender = "F"}(customer_customer_id = customer_id \wedge trans_id = trans_id)$

```

1 • SELECT RS.Customer.first_name AS Customer_firstname, RS.Customer.last_name AS Customer_lastname,
2     RS.Driver.first_name AS Driver_firstname, RS.Driver.last_name AS Driver_lastname FROM RS.Customer
3     JOIN RS.Trans ON Trans.Customer_id = Customer.customer_id
4     JOIN RS.Trans_data ON Trans_data.transaction_id = Trans.transaction_id
5     JOIN RS.Cars ON Cars.car_id = Trans.car_id
6     JOIN RS.Driver ON Driver.driver_id = Cars.driver_id
7     WHERE Trans.transaction_date BETWEEN '2022-11-22' AND '2022-12-26' AND Customer.gender = 'F'
8     ORDER BY RS.Customer.first_name;
```

Result Grid				
Filter Rows: <input type="text"/>				
Export: <input type="button" value=""/>				
Wrap Cell Content: <input type="button" value=""/>				
	Customer_firstname	Customer_lastname	Driver_firstname	Driver_lastname
▶	Éles	Barbara	Sebastian	Vettel
	Éles	Barbara	Lakatos	Bence
	Kis	Heni	Sebastian	Vettel
	Kopasz	Dominika	Michael	Schumacher
	Kopasz	Dominika	Budai	Nikoletta
	Kovács	Dorina	Sebastian	Vettel
	Nagy	Lidia	Budai	Nikoletta
	Nagy	Lidia	Vass	Gábor
	Pánkusi	Barbara	Fülöp	Dalma