TALLINNA TÖÖSTUSHARIDUSKESKUS

„Auto Rent Database“  
Aruanne andmebaas

Õpilane: Darja Suhhanova  
Õpetaja: Irina Merkulova

Tallinn  
2023

**Contents:**

[Subject: 3](#_Toc167275439)

[Objects: 3](#_Toc167275440)

[Actions: 3](#_Toc167275441)

[Goals: 3](#_Toc167275442)

[Data Schema: 3](#_Toc167275443)

[SQL queries for creating tables 4](#_Toc167275444)

[Filled tables – screenshots 6](#_Toc167275445)

[Show related tables using SELECT …. INNER JOIN 7](#_Toc167275446)

[Diagram of related tables from the server 8](#_Toc167275447)

[Users and their permissions 9](#_Toc167275448)

[Checking user activity 9](#_Toc167275449)

[Creating triggers 10](#_Toc167275450)

[Triggers for Car table 12](#_Toc167275451)

[Triggers for Client table 13](#_Toc167275452)

[Triggers for Agreement table 14](#_Toc167275453)

[Procedure: 16](#_Toc167275454)

[Table 1: Users and their permissions 8](#_Toc167221608)

# Subject:

AutoRent

# Objects:

Client, employee, car\_model, car\_make, car\_type, car\_class, car, agreement (contract), rent duration,

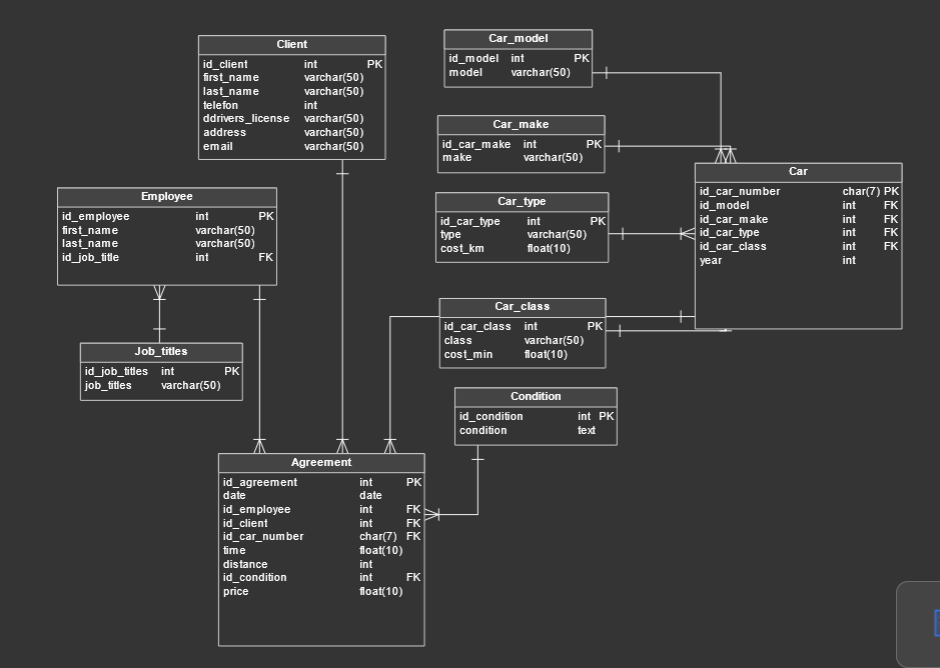
# Actions:

A car has a make, model, type, and class.  
A car can be rented, where the rental has a duration and distance traveled.  
A client rents a car.  
An employee fills in the contract details and notes the car's condition after the deal is completed.

# Goals:

1. Management of clients
2. Management of cars
3. Management of contracts and condition of cars

# Data Schema:



Picture Data Schema

# SQL queries for creating tables

-- Table: Agreement

CREATE TABLE Agreement (

id\_agreement int NOT NULL,

date\_ date NOT NULL,

id\_employee int NOT NULL,

id\_client int NOT NULL,

id\_car\_number char(7) NOT NULL,

time\_ float(10) NOT NULL,

distance int NOT NULL,

id\_condition int NOT NULL,

price float(10) NOT NULL,

CONSTRAINT Agreement\_pk PRIMARY KEY (id\_agreement)

);

-- Table: Car

CREATE TABLE Car (

id\_car\_number char(7) NOT NULL,

id\_model int NOT NULL,

id\_car\_make int NOT NULL,

id\_car\_type int NOT NULL,

id\_car\_class int NOT NULL,

car\_year int NOT NULL,

CONSTRAINT Car\_pk PRIMARY KEY (id\_car\_number)

);

-- Table: Car\_class

CREATE TABLE Car\_class (

id\_car\_class int NOT NULL,

class varchar(50) NOT NULL,

cost\_min float(10) NOT NULL,

CONSTRAINT Car\_class\_pk PRIMARY KEY (id\_car\_class)

);

-- Table: Car\_make

CREATE TABLE Car\_make (

id\_car\_make int NOT NULL,

make varchar(50) NOT NULL,

CONSTRAINT Car\_make\_pk PRIMARY KEY (id\_car\_make)

);

-- Table: Car\_model

CREATE TABLE Car\_model (

id\_model int NOT NULL,

model varchar(50) NOT NULL,

CONSTRAINT Car\_model\_pk PRIMARY KEY (id\_model)

);

-- Table: Car\_type

CREATE TABLE Car\_type (

id\_car\_type int NOT NULL,

car\_type varchar(50) NOT NULL,

cost\_km float(10) NOT NULL,

CONSTRAINT Car\_type\_pk PRIMARY KEY (id\_car\_type)

);

-- Table: Client

CREATE TABLE Client (

id\_client int NOT NULL,

first\_name varchar(50) NOT NULL,

last\_name varchar(50) NOT NULL,

telefon int NOT NULL,

drivers\_license varchar(50) NOT NULL,

client\_address varchar(50) NOT NULL,

email varchar(50) NOT NULL,

CONSTRAINT Client\_pk PRIMARY KEY (id\_client)

);

-- Table: Condition

CREATE TABLE Condition (

id\_condition int NOT NULL,

condition text NOT NULL,

CONSTRAINT Condition\_pk PRIMARY KEY (id\_condition)

);

-- Table: Employee

CREATE TABLE Employee (

id\_employee int NOT NULL,

first\_name varchar(50) NOT NULL,

last\_name varchar(50) NOT NULL,

id\_job\_title int NOT NULL,

CONSTRAINT Employee\_pk PRIMARY KEY (id\_employee)

);

-- Table: Job\_titles

CREATE TABLE Job\_title (

id\_job\_title int NOT NULL,

job\_title varchar(50) NOT NULL,

CONSTRAINT Job\_titles\_pk PRIMARY KEY (id\_job\_title)

);

-- foreign keys

ALTER TABLE Agreement ADD CONSTRAINT Agreement\_Car

FOREIGN KEY (id\_car\_number)

REFERENCES Car (id\_car\_number);

-- Reference: Agreement\_Client (table: Agreement)

ALTER TABLE Agreement ADD CONSTRAINT Agreement\_Client

FOREIGN KEY (id\_client)

REFERENCES Client (id\_client);

-- Reference: Agreement\_Condition (table: Agreement)

ALTER TABLE Agreement ADD CONSTRAINT Agreement\_Condition

FOREIGN KEY (id\_condition)

REFERENCES Condition (id\_condition);

-- Reference: Agreement\_Employee (table: Agreement)

ALTER TABLE Agreement ADD CONSTRAINT Agreement\_Employee

FOREIGN KEY (id\_employee)

REFERENCES Employee (id\_employee);

-- Reference: Car\_Car\_class (table: Car)

ALTER TABLE Car ADD CONSTRAINT Car\_Car\_class

FOREIGN KEY (id\_car\_class)

REFERENCES Car\_class (id\_car\_class);

-- Reference: Car\_Car\_make (table: Car)

ALTER TABLE Car ADD CONSTRAINT Car\_Car\_make

FOREIGN KEY (id\_car\_make)

REFERENCES Car\_make (id\_car\_make);

-- Reference: Car\_Car\_model (table: Car)

ALTER TABLE Car ADD CONSTRAINT Car\_Car\_model

FOREIGN KEY (id\_model)

REFERENCES Car\_model (id\_model);

-- Reference: Car\_Car\_type (table: Car)

ALTER TABLE Car ADD CONSTRAINT Car\_Car\_type

FOREIGN KEY (id\_car\_type)

REFERENCES Car\_type (id\_car\_type);

-- Reference: Employee\_Job\_titles (table: Employee)

ALTER TABLE Employee ADD CONSTRAINT Employee\_Job\_title

FOREIGN KEY (id\_job\_title)

REFERENCES Job\_title (id\_job\_title);

# Filled tables – screenshots

A screenshot of a computer

Description automatically generated

Picture Table: car\_class

A screenshot of a computer

Description automatically generated

Picture Table: car\_make

A screenshot of a computer

Description automatically generated

Picture Table: car\_model

A screenshot of a computer

Description automatically generated

Picture Table: car\_type

A screenshot of a computer

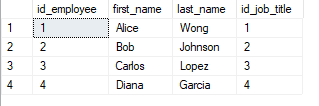
Description automatically generated

Picture Table: Client

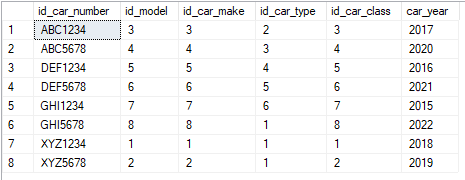
A screenshot of a computer

Description automatically generated

Picture Table: Job\_titles



Picture Table: Employee



Picture Table: Car

A screenshot of a computer

Description automatically generated

Picture Table: Agreement

A screenshot of a computer

Description automatically generated

Picture Table: Condition

# Show related tables using SELECT …. INNER JOIN

A screenshot of a computer

Description automatically generated

Picture : related tables

SELECT

a.id\_agreement,

a.date\_,

e.last\_name as Employee,

cl.last\_name as ClientLastName,

a.id\_car\_number as CarNumber,

a.time\_ AS TimeH,

a.distance AS DistanceKM,

co.condition as ConditionRank,

(a.distance \* ct.cost\_km) + (a.time\_ \* cc.cost\_min) AS calculated\_price

FROM

Agreement a

INNER JOIN Car c ON a.id\_car\_number = c.id\_car\_number

INNER JOIN Car\_type ct ON c.id\_car\_type = ct.id\_car\_type

INNER JOIN Car\_class cc ON c.id\_car\_class = cc.id\_car\_class

INNER JOIN Employee e ON a.id\_employee = e.id\_employee

INNER JOIN Client cl ON a.id\_client = cl.id\_client

INNER JOIN Condition co ON a.id\_condition = co.id\_condition;

# Diagram of related tables from the server

A screenshot of a computer

Description automatically generated

Picture : Diagram of related tables

# 

# Users and their permissions

Director – Full permissions (SELECT, UPDATE, INSERT) on all tables.

Salesman – SELECT and INSERT permissions on specific tables.

Manager Full permissions (SELECT, UPDATE, INSERT) on specific tables.

|  |  |  |  |
| --- | --- | --- | --- |
| Tabel | director (admin) | manager | salesman |
| Agreement | SELECT, INSERT, UPDATE, DELETE | SELECT, UPDATE, INSERT | SELECT, INSERT |
| Client | SELECT, INSERT, UPDATE, DELETE | SELECT, UPDATE, INSERT | SELECT, INSERT |
| Car | SELECT, INSERT, UPDATE, DELETE | SELECT, UPDATE, INSERT | SELECT |
| Condition | SELECT, INSERT, UPDATE, DELETE |  |  |
| Car\_class | SELECT, INSERT, UPDATE, DELETE |  |  |
| Car\_model | SELECT, INSERT, UPDATE, DELETE |  |  |
| Car\_make | SELECT, INSERT, UPDATE, DELETE |  |  |
| Car\_type | SELECT, INSERT, UPDATE, DELETE |  |  |
| logi | SELECT, INSERT, UPDATE, DELETE |  |  |

Table : Users and their permissions

GRANT SELECT, UPDATE, INSERT ON andmebaasinimi.Laps TO tootaja

# Checking user activity

# Creating triggers

1. A screenshot of a computer

   Description automatically generatedCreate triggers to track user actions.

Picture Triggers: Car, Agreement, Client

1. Create one table named **logi** that records the activity of all created triggers on all important (related) tables and can be viewed only by the admin (in my case director).

**A screenshot of a computer

Description automatically generated**

Picture 15 Triggers: Logi is visible only to the director

In this table **logi**, the following is displayed:

1. **Date/time** (insertions, deletions, and updates of records in some important tables (at least two tables)),
2. the username of the user who performed the action (using the function **USER()**)
3. A screenshot of a computer

   Description automatically generatedand all data from the affected table using the function **concat()**

Picture Triggers: Table 'logi'

Triggers for Car table

A screenshot of a computer program

Description automatically generated

Picture Trigger: Car\_INSERT

A close-up of a text

Description automatically generated

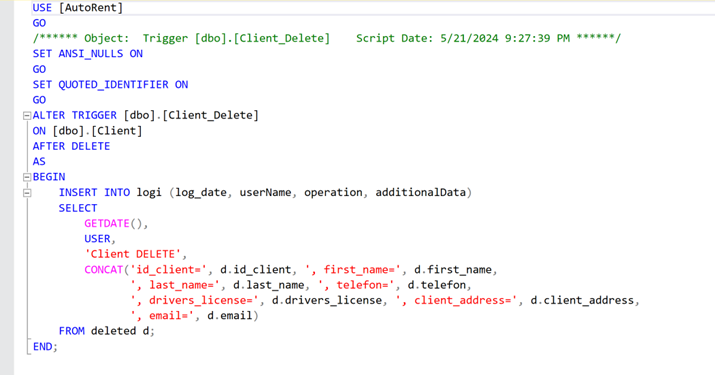
Picture Trigger: Car\_UPDATE

A screenshot of a computer program

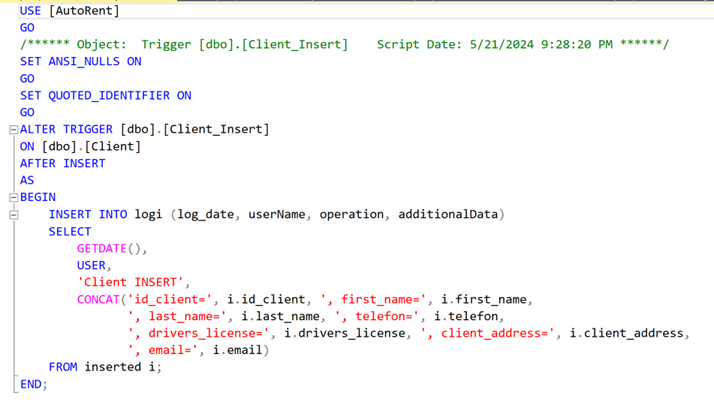
Description automatically generated

Picture Trigger: Car\_DELETE

Triggers for Client table



Picture Trigger: Client\_DELETE



Picture Trigger: Client\_INSERT

A screenshot of a computer program

Description automatically generated

Picture Trigger: Client\_UPDATE

## Triggers for Agreement table

A computer screen shot of a program

Description automatically generated

Picture Trigger: Agreement\_DELETE

**A screen shot of a computer code

Description automatically generated**

Picture Trigger: Agreement\_INSERT

**A screenshot of a computer program

Description automatically generated**

Picture Trigger: Agreement\_UPDATE

create trigger ClientRentsCar

ON Agreement

AFTER INSERT

AS

BEGIN

INSERT INTO logi (log\_date, userName, operation, additionalData)

SELECT

GETDATE(),

USER,

'Client rents a car!',

CONCAT('id\_agreement=', i.id\_agreement, c.first\_name, ' ', c.last\_name,

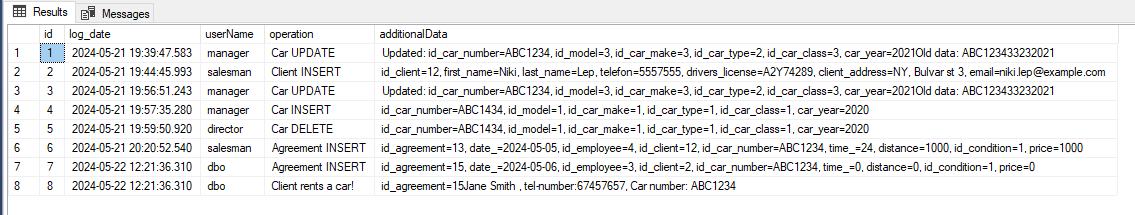
' , tel-number:', c.telefon, ', Car number: ', i.id\_car\_number)

FROM inserted i

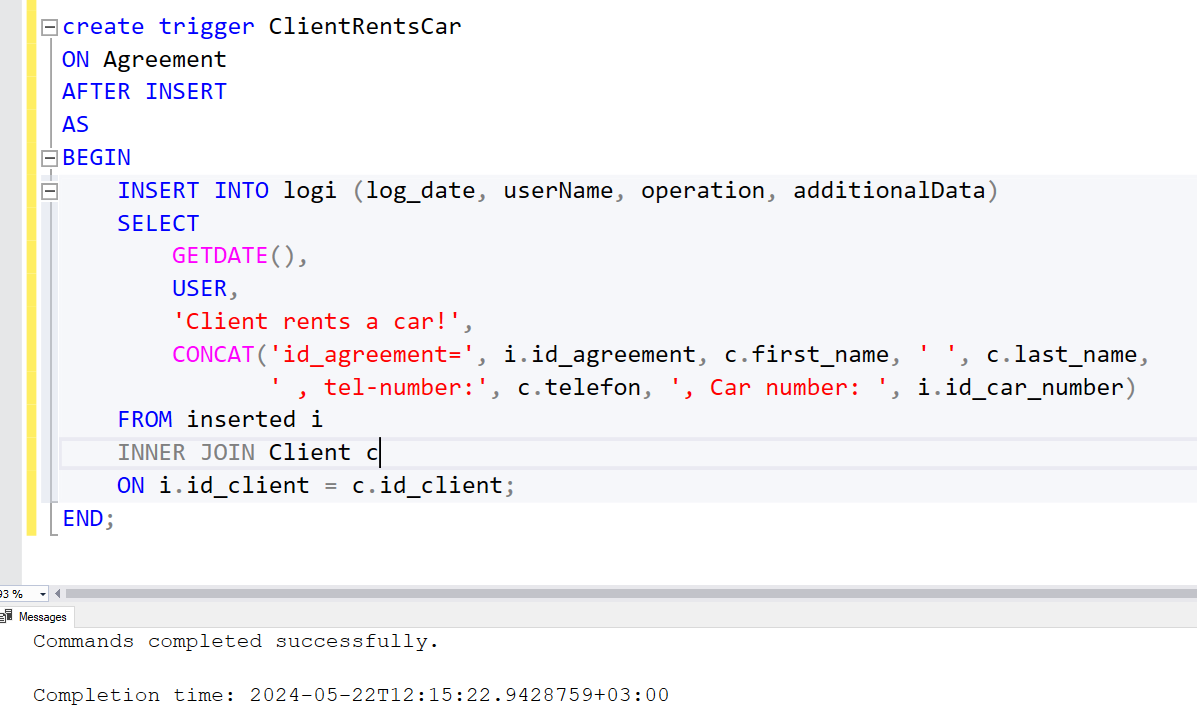
INNER JOIN Client c

ON i.id\_client = c.id\_client;

END;

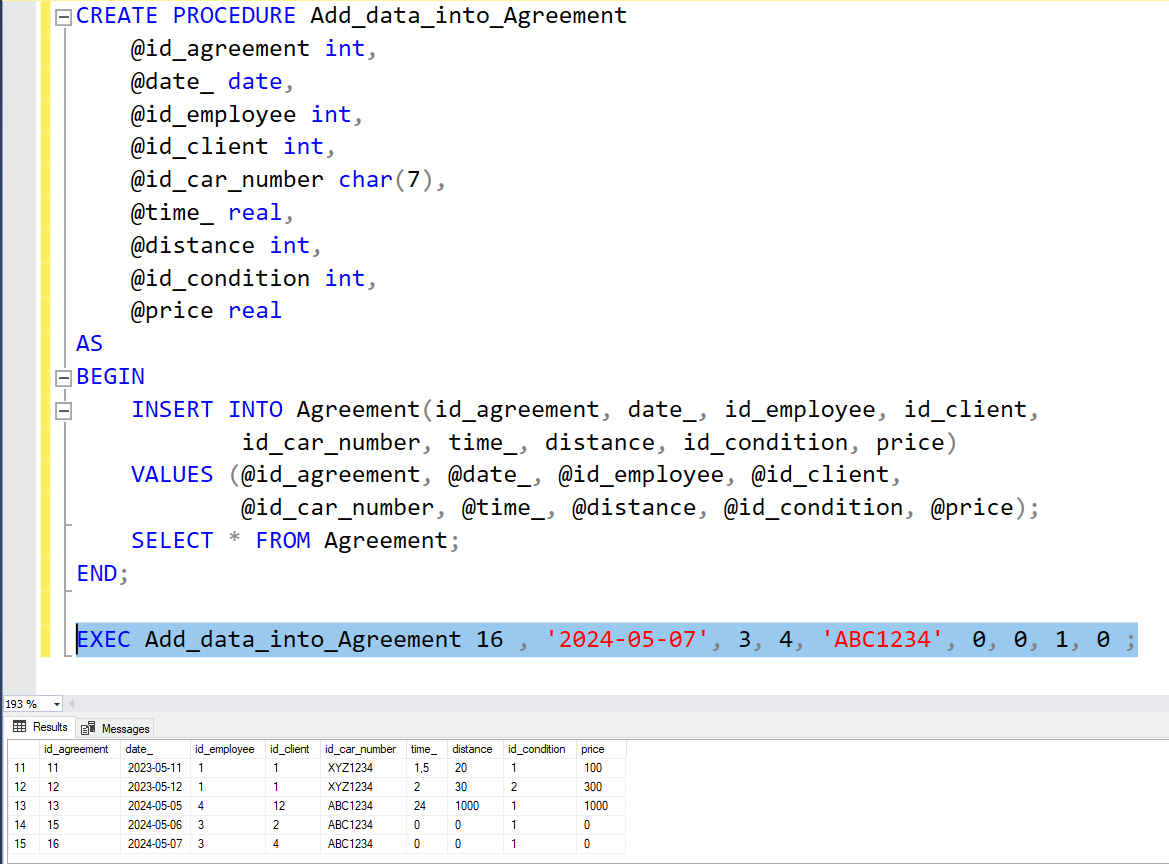


Picture 26. : Client rents a car



Picture 26. Trigger: Client rents a Car

# Procedure:



Picture 27 Procedure: Add\_data\_into\_Agreement

CREATE PROCEDURE Add\_data\_into\_Agreement

@id\_agreement int,

@date\_ date,

@id\_employee int,

@id\_client int,

@id\_car\_number char(7),

@time\_ real,

@distance int,

@id\_condition int,

@price real

AS

BEGIN

INSERT INTO Agreement(id\_agreement, date\_, id\_employee, id\_client,

id\_car\_number, time\_, distance, id\_condition, price)

VALUES (@id\_agreement, @date\_, @id\_employee, @id\_client,

@id\_car\_number, @time\_, @distance, @id\_condition, @price);

SELECT \* FROM Agreement;

END;