Предметная область: «Прокат автомобилей»

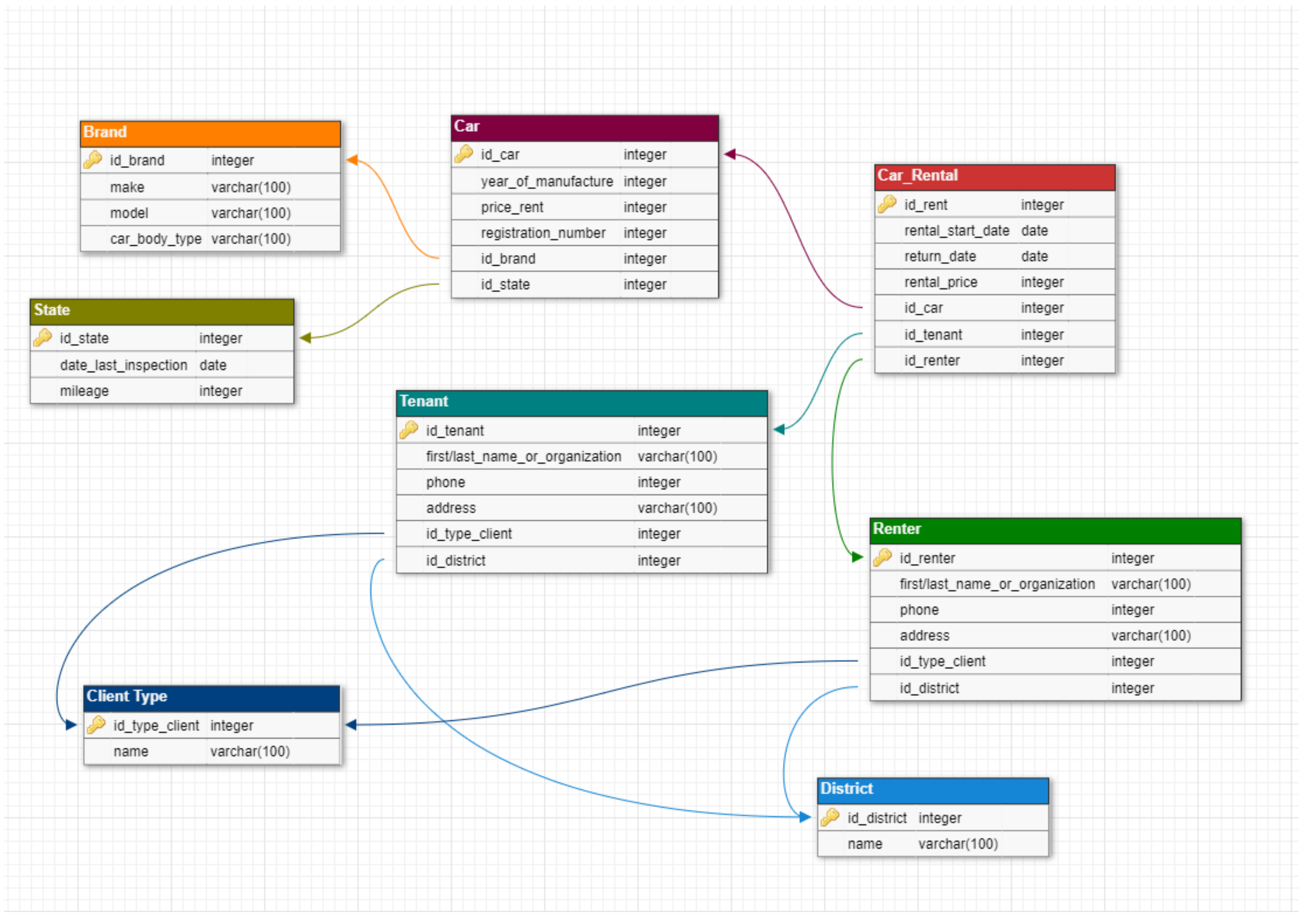
Реализация: MS SQLServer

Оглавление

ОПИСАНИЕ СИСТЕМЫ
Требования

Разработайте структуру базы данных, на основе которой можно будет создать приложение для ведения реестра договоров аренды.
Договор аренды – это документ, регулирующий отношения между арендатором и арендодателем. В договоре определяется объект аренды (земельный участок, имеющий адрес, тип и описание), а также периоды, суммы и сроки оплат. Арендаторами и арендодателями могут быть органы власти, а также юридические и физические лица. Договор заключается на определенный срок между арендатором и одним или более арендодателями. Более одного субъекта на стороне арендодателя может быть, только если договор заключается с физическими лицами.

Модель данных



Функциональность

Серверная часть

Процедуры:

1. AddNewTenant — добавление арендатора
2.CalculateCost —суммарная стоимость проката за период для арендатора
3.CheckInspection — получить список тех автомобилей, которые 3 года с текущей даты не были на техосмотре.
4.Удаление заказа
5. Получение количества заказов арендатора

Триггеры:
1.Tenant_INSERT — сохранение в истрии события добавления сотрудника
2.Tenant_Check — проверка добавления арендатора, Иванов Иван не может быть им.
3.Renter_Address_UPDATE – сохранение в истории события обновления адреса арендодателя.
4. Запрет на удаление арендодателей, являющихся юр.лицами
5. Контроль повторного добавления автомобиля

Представления:
1.V_District_Dorset - заказы арендаторов из района Dorset
2.V_Physical_Person - список заказов у арендодателей, являющихся физическими лицами.
3.V_Order_List - список заказов с именами арендаторов и арендодателей
4.Получение списка автомобилей и их марок
5. Список заказов с данными автомобиля

Клиентская часть

(1) Добавление префикса к адресу арендодателя с ID района равным 2 или 3
(2)Добавляем префикс к регистрационному номеру машины, если пробег автомобиля больше 100 000 км
(3)Увеличение цены аренды автомобилей на 200 рублей, если последний техосмотр был после 2020 года
(4)Удаление арендаторов, которые сделали заказ на аренду больше месяца
(5)Максимальный и минимальный цены заказов среди физ.лиц
(6)Группировка машин по количеству заказов, среди тех, которые заказывали хотя бы один раз
(7)Арендаторы, отсортированные по дате начала аренды
(8)Определение к какому типу клиентов относятся арендодатели
(9)Процент заказов из района Dorset
(10)Поиск всех арендаторов и арендодателей, которые заказывали/предоставляли машину весной
(11)Данные машин, у которых дата последнего техосмотра была до 2020 года
(12)Имя арендатора и стоимость аренды за указанный период

СКРИПТЫ

```
--------------------------------------------
-- Init
--------------------------------------------
CREATE DATABASE rent_db;
GO
USE rent_db;
--------------------------------------------
-- TABLE
--------------------------------------------
CREATE TABLE District(
   District_ID                                   INTEGER        NOT NULL,
   Name                                          VARCHAR(100)    NOT NULL,
CONSTRAINT District_PK PRIMARY KEY (District_ID)
)
;
CREATE TABLE Client_Type(
   Client_Type_ID            INTEGER        NOT NULL,
   Name                                          VARCHAR(100)    NOT NULL,
CONSTRAINT Client_Type_PK PRIMARY KEY (Client_Type_ID)
)
;
CREATE TABLE Condition(
   Condition_ID              INTEGER        NOT NULL,
   Date_Last_Inspection            DATE DEFAULT GETDATE() NOT NULL,
   Mileage                        INTEGER              NOT NULL,
CONSTRAINT Condition_PK PRIMARY KEY (Condition_ID)
)
;

CREATE TABLE Brand(
   Brand_ID                 INTEGER        NOT NULL,
   make                  VARCHAR(100),
        model                                     VARCHAR(100),
CONSTRAINT Brand_PK PRIMARY KEY (Brand_ID)
)
;
CREATE TABLE Renter(
   Renter_ID          INTEGER      NOT NULL,
   Full_Name_or_Organization     VARCHAR(100),
   Phone                                         INTEGER        NOT NULL,
        Address                           VARCHAR(100),
        Client_Type_ID                           INTEGER         NOT NULL,
        District_ID                              INTEGER         NOT NULL,
CONSTRAINT Renter_PK PRIMARY KEY (Renter_ID)
)
;
```

```
CREATE TABLE Tenant(
   Tenant_ID                                    INTEGER        NOT NULL,
   Full_Name_or_Organization     VARCHAR(100),
   Phone                                        INTEGER        NOT NULL,
        Address                        VARCHAR(100),
        Client_Type_ID                          INTEGER,
        District_ID                             INTEGER,
CONSTRAINT Tenant_PK PRIMARY KEY (Tenant_ID)
)
;
CREATE TABLE Car(
   Car_ID                                       INTEGER      NOT NULL,
   Year_Of_Manufacture                 INTEGER  NOT NULL,
   Price                                        INTEGER,
   Registration_Number                 INTEGER  NOT NULL,
        Brand_ID                                INTEGER         NOT NULL,
        Condition_ID                            INTEGER         NOT NULL,
CONSTRAINT Car_PK PRIMARY KEY (Car_ID)
)
;
CREATE TABLE Car_Rental(
   Rent_ID                                      INTEGER        NOT NULL,
   Rental_Start_Date                   DATE        DEFAULT GETDATE()     NOT
NULL,
   Return_Date                                  DATE     DEFAULT GETDATE()
NOT NULL,
        Car_ID                                          INTEGER          NOT NULL,
        Tenant_ID                               INTEGER         NOT NULL,
        Renter_ID                               INTEGER         NOT NULL,
CONSTRAINT Rent_PK PRIMARY KEY (Rent_ID)
)
;

CREATE TABLE History(
        History_ID                    INT IDENTITY PRIMARY KEY,
        Operation                          VARCHAR(100) ,
        CreateAt                           DATETIME DEFAULT GETDATE(),
)
;
--------------------------------------------
-- FOREIGN KEY
--------------------------------------------

ALTER TABLE Car ADD CONSTRAINT FK_Car_Brand
   FOREIGN KEY (Brand_ID)
   REFERENCES Brand(Brand_ID)
;
```

```
ALTER TABLE Car ADD CONSTRAINT FK_Car_Condition
   FOREIGN KEY (Condition_ID)
   REFERENCES Condition(Condition_ID)
;
ALTER TABLE Tenant ADD CONSTRAINT FK_Tenant_Client_Type
   FOREIGN KEY (Client_Type_ID)
   REFERENCES Client_Type(Client_Type_ID)
;
ALTER TABLE Tenant ADD CONSTRAINT FK_Tenant_District
   FOREIGN KEY (District_ID)
   REFERENCES District(District_ID)
;
ALTER TABLE Renter ADD CONSTRAINT FK_Renter_Client_Type
   FOREIGN KEY (Client_Type_ID)
   REFERENCES Client_Type(Client_Type_ID)
;
ALTER TABLE Renter ADD CONSTRAINT FK_Renter_District
   FOREIGN KEY (District_ID)
   REFERENCES District(District_ID)
;
ALTER TABLE Car_Rental ADD CONSTRAINT FK_Car_Rental_Car
   FOREIGN KEY (Car_ID)
   REFERENCES Car(Car_ID)
;
ALTER TABLE Car_Rental ADD CONSTRAINT FK_Car_Rental_Tenant
   FOREIGN KEY (Tenant_ID)
   REFERENCES Tenant(Tenant_ID)
;
ALTER TABLE Car_Rental ADD CONSTRAINT FK_Car_Rental_Renter
   FOREIGN KEY (Renter_ID)
   REFERENCES Renter(Renter_ID)
;


-----------------------------------------------------------------
-- Заполнение таблиц тестовыми данными
-----------------------------------------------------------------
INSERT INTO District(District_ID, Name) VALUES (1, 'Cornwall');
INSERT INTO District(District_ID, Name) VALUES (2, 'Dorset');
INSERT INTO District(District_ID, Name) VALUES (3, 'Breckland');
INSERT INTO District(District_ID, Name) VALUES (4, 'Ryedale');
INSERT INTO District(District_ID, Name) VALUES (5, 'Lakeland');
INSERT INTO District(District_ID, Name) VALUES (6, 'Allerdale');


INSERT INTO Client_Type(Client_Type_ID, Name) VALUES (1,'legal person');
INSERT INTO Client_Type(Client_Type_ID, Name) VALUES (2, 'physical person');
INSERT INTO Client_Type(Client_Type_ID, Name) VALUES (3,'government');
INSERT INTO Condition(Condition_ID, Date_Last_Inspection, Mileage) VALUES
(1,'2018-06-09', 100000);
INSERT INTO Condition(Condition_ID, Date_Last_Inspection, Mileage) VALUES
(2,'2020-07-02', 215000);
INSERT INTO Condition(Condition_ID, Date_Last_Inspection, Mileage) VALUES
(3,'2019-03-01', 70000);
INSERT INTO Condition(Condition_ID, Date_Last_Inspection, Mileage) VALUES
(4,'2019-09-12', 164000);
INSERT INTO Condition(Condition_ID, Date_Last_Inspection, Mileage) VALUES
(5,'2020-06-17', 320000);
INSERT INTO Condition(Condition_ID, Date_Last_Inspection, Mileage) VALUES
(6,'2020-08-04', 10000);
INSERT INTO Condition(Condition_ID, Date_Last_Inspection, Mileage) VALUES
(7,'2019-04-03', 244000);
INSERT INTO Condition(Condition_ID, Date_Last_Inspection, Mileage) VALUES
(8,'2019-03-02', 186000);

INSERT INTO Brand(Brand_ID, make, model) VALUES (1, 'Skoda', 'Octavia');
INSERT INTO Brand(Brand_ID, make, model) VALUES (2, 'Subaru', 'Forester');
INSERT INTO Brand(Brand_ID, make, model) VALUES (3, 'Volkswagen',
'Transporter');
INSERT INTO Brand(Brand_ID, make, model) VALUES (4, 'Lada', 'Vesta');
INSERT INTO Brand(Brand_ID, make, model) VALUES (5, 'Skoda','Yeti');
INSERT INTO Brand(Brand_ID, make, model) VALUES (6, 'Lada', 'Granta');
INSERT INTO Brand(Brand_ID, make, model) VALUES (7, 'Volkswaggen', 'Polo');
INSERT INTO Brand(Brand_ID, make, model) VALUES (8, 'Volkswagen', 'Tiguan');
INSERT INTO Brand(Brand_ID, make, model) VALUES (9, 'Toyota', 'Kamry');
INSERT INTO Brand(Brand_ID, make, model) VALUES (10, 'Toyota', 'Corolla');


INSERT INTO Renter(Renter_ID, Full_Name_or_Organization, Phone, Address,
Client_Type_ID, District_ID) VALUES (1, 'John Smith', 77000, 'Rudy Forest, 32', 2,
5);
INSERT INTO Renter(Renter_ID, Full_Name_or_Organization, Phone, Address,
Client_Type_ID, District_ID) VALUES (2, 'Alan Green', 79180, 'McCullough Trace,
44', 2, 6);
INSERT INTO Renter(Renter_ID, Full_Name_or_Organization, Phone, Address,
Client_Type_ID, District_ID) VALUES (3, 'Peter Mackenzie', 136234, 'Dean Mount,
114', 2, 4);
INSERT INTO Renter(Renter_ID, Full_Name_or_Organization, Phone, Address,
Client_Type_ID, District_ID) VALUES (4, 'Mary Higgins', 770177, 'Oran Village,
76', 2, 1);
INSERT INTO Renter(Renter_ID, Full_Name_or_Organization, Phone, Address,
Client_Type_ID, District_ID) VALUES (5, 'Ltd. Builder', 772311, 'Evelyn Turnpike,
157', 1, 2);
INSERT INTO Renter(Renter_ID, Full_Name_or_Organization, Phone, Address,
```

Client_Type_ID, District_ID) VALUES (6, 'Ltd. Nature', 750345, 'Kelli Fords, 23', 1, 3);
INSERT INTO Renter(Renter_ID, Full_Name_or_Organization, Phone, Address, Client_Type_ID, District_ID) VALUES (7, 'Ltd. Development', 280065, 'Haylee Haven, 81', 1, 5);
INSERT INTO Renter(Renter_ID, Full_Name_or_Organization, Phone, Address, Client_Type_ID, District_ID) VALUES (8, 'Anne Smith', 7700168, 'Ima Walks, 109', 2, 6);

INSERT INTO Tenant(Tenant_ID, Full_Name_or_Organization, Phone, Address, Client_Type_ID, District_ID) VALUES (1, 'Ltd. Traveler’s Tale', 774848, 'McCullough, 12', 1, 3);
INSERT INTO Tenant(Tenant_ID, Full_Name_or_Organization, Phone, Address, Client_Type_ID, District_ID) VALUES (2, 'Ltd. Best Buyer', 745108, 'Melvina Plains, 27', 1, 2);
INSERT INTO Tenant(Tenant_ID, Full_Name_or_Organization, Phone, Address, Client_Type_ID, District_ID) VALUES (3, 'Eloise Turner', 774112, 'Cale Extensions, 79', 2, 2);
INSERT INTO Tenant(Tenant_ID, Full_Name_or_Organization, Phone, Address, Client_Type_ID, District_ID) VALUES (4, 'Erik Bolton', 270076, 'Crist Route, 178', 2, 2);
INSERT INTO Tenant(Tenant_ID, Full_Name_or_Organization, Phone, Address, Client_Type_ID, District_ID) VALUES (5, 'Marie Ward', 756102, 'Emely Plains, 64', 2, 1);
INSERT INTO Tenant(Tenant_ID, Full_Name_or_Organization, Phone, Address, Client_Type_ID, District_ID) VALUES (6, 'Arabella Ryan', 767281, 'Frami Ford, 70', 2, 2);
INSERT INTO Tenant(Tenant_ID, Full_Name_or_Organization, Phone, Address, Client_Type_ID, District_ID) VALUES (7, 'Fergus Daniel', 783909, 'Moses Tunnel, 53', 2, 3);
INSERT INTO Tenant(Tenant_ID, Full_Name_or_Organization, Phone, Address, Client_Type_ID, District_ID) VALUES (8, 'Fergu Mikle', 783789, 'Moses Tunnel, 72', 2, 5);

INSERT INTO Car(Car_ID, Year_Of_Manufacture, Price, Registration_Number, Brand_ID, Condition_ID) VALUES (1, 2004, 2600, 777, 1, 4);
INSERT INTO Car(Car_ID, Year_Of_Manufacture, Price, Registration_Number, Brand_ID, Condition_ID) VALUES (2, 2006, 3000, 895, 1, 5);
INSERT INTO Car(Car_ID, Year_Of_Manufacture, Price, Registration_Number, Brand_ID, Condition_ID) VALUES (3, 2017, 2000, 124, 4, 1);
INSERT INTO Car(Car_ID, Year_Of_Manufacture, Price, Registration_Number, Brand_ID, Condition_ID) VALUES (4, 2014, 1700, 768, 1, 2);
INSERT INTO Car(Car_ID, Year_Of_Manufacture, Price, Registration_Number, Brand_ID, Condition_ID) VALUES (5, 2001, 3500, 786, 3, 7);
INSERT INTO Car(Car_ID, Year_Of_Manufacture, Price, Registration_Number, Brand_ID, Condition_ID) VALUES (6, 2007, 3200, 465, 5, 5);
INSERT INTO Car(Car_ID, Year_Of_Manufacture, Price, Registration_Number,

```
Brand_ID, Condition_ID) VALUES (7, 2020, 2500, 098, 6, 2);
INSERT INTO Car(Car_ID, Year_Of_Manufacture, Price, Registration_Number,
Brand_ID, Condition_ID) VALUES (8, 2014, 2300, 659, 10, 6);
INSERT INTO Car(Car_ID, Year_Of_Manufacture, Price, Registration_Number,
Brand_ID, Condition_ID) VALUES (9, 2009, 2600, 868, 8, 8);
INSERT INTO Car(Car_ID, Year_Of_Manufacture, Price, Registration_Number,
Brand_ID, Condition_ID) VALUES (10, 2015, 2700, 741, 7, 1);

INSERT INTO Car_Rental(Rent_ID, Rental_Start_Date, Return_Date, Car_ID,
Tenant_ID, Renter_ID) VALUES (1, '2020-01-04', '2020-01-07', 2, 1, 1);
INSERT INTO Car_Rental(Rent_ID, Rental_Start_Date, Return_Date, Car_ID,
Tenant_ID, Renter_ID) VALUES (2, '2020-01-07', '2020-01-12', 3, 2, 2);
INSERT INTO Car_Rental(Rent_ID, Rental_Start_Date, Return_Date, Car_ID,
Tenant_ID, Renter_ID) VALUES (3, '2020-07-17', '2020-08-01', 5, 3, 4);
INSERT INTO Car_Rental(Rent_ID, Rental_Start_Date, Return_Date, Car_ID,
Tenant_ID, Renter_ID) VALUES (4, '2020-08-30', '2020-09-02', 4, 4, 3);
INSERT INTO Car_Rental(Rent_ID, Rental_Start_Date, Return_Date, Car_ID,
Tenant_ID, Renter_ID) VALUES (5, '2020-08-15', '2020-08-19', 7, 5, 6);
INSERT INTO Car_Rental(Rent_ID, Rental_Start_Date, Return_Date, Car_ID,
Tenant_ID, Renter_ID) VALUES (6, '2021-11-04', '2021-11-17', 1, 6, 7);
INSERT INTO Car_Rental(Rent_ID, Rental_Start_Date, Return_Date, Car_ID,
Tenant_ID, Renter_ID) VALUES (7, '2021-10-17', '2021-10-24', 1, 7, 8);
INSERT INTO Car_Rental(Rent_ID, Rental_Start_Date, Return_Date, Car_ID,
Tenant_ID, Renter_ID) VALUES (8, '2021-02-14', '2021-02-23', 8, 1, 5);
INSERT INTO Car_Rental(Rent_ID, Rental_Start_Date, Return_Date, Car_ID,
Tenant_ID, Renter_ID) VALUES (9, '2021-04-09', '2021-04-11', 9, 4, 3);
INSERT INTO Car_Rental(Rent_ID, Rental_Start_Date, Return_Date, Car_ID,
Tenant_ID, Renter_ID) VALUES (10, '2021-05-09', '2021-05-12', 2, 6, 1);

--------------------------------------------
-- INDEX
--------------------------------------------

CREATE INDEX idx_brand_name ON Brand(Make, Model);
CREATE INDEX idx_car_registr_number ON Car(Registration_Number);
CREATE INDEX idx_car_rental_period ON Car_Rental(Rental_Start_Date,
Return_Date);

--------------------------------------------
-- PROCEDURE, FUNCTION
--------------------------------------------

GO
CREATE PROCEDURE AddNewTenant(
@var_full_name VARCHAR(100),
@var_phone INTEGER,
@var_address VARCHAR(100),
@var_type INTEGER,
@var_district INTEGER)
AS
BEGIN
  DECLARE @var_new_tenant_id INTEGER;
  SELECT @var_new_tenant_id = MAX(Tenant.Tenant_ID) + 1 FROM Tenant;
  INSERT INTO Tenant(Tenant_ID, Full_Name_or_Organization, Phone, Address,
Client_Type_ID, District_ID)
      VALUES(@var_new_tenant_id, @var_full_name, @var_phone, @var_address,
@var_type, @var_district);
END;
--EXECUTE AddNewTenant 'Ivan Ivanov', 48595, 'Sadovaya, 3', 2, 4;
--SELECT * FROM Tenant;


GO
CREATE FUNCTION CalculateCost(@tenant_id INTEGER) RETURNS INTEGER
BEGIN
        DECLARE @cost INTEGER;
        DECLARE @diff INTEGER;
        SELECT @diff = DATEDIFF(day, Car_Rental.Rental_Start_Date,
Car_Rental.Return_Date)
        FROM Car_Rental WHERE Car_Rental.Tenant_ID = @tenant_id;

        SELECT @cost = @diff*Car.Price FROM Car, Car_Rental WHERE
Car_Rental.Tenant_ID = @tenant_id
        AND Car.Car_ID =  Car_Rental.Car_ID;
        RETURN @cost;
END;
GO

--SELECT dbo.CalculateCost(1);

GO
CREATE FUNCTION CheckInspection(@cur_date DATE) RETURNS TABLE AS
RETURN
(
        SELECT Car.Car_ID, Car.Registration_Number,
Condition.Date_Last_Inspection
                FROM Car, Condition
                WHERE Car.Condition_ID = Condition.Condition_ID AND
                DATEDIFF(year, Condition.Date_Last_Inspection, @cur_date) > 2
);
GO
--SELECT * FROM dbo.CheckInspection(CONVERT(DATE, GETDATE()));

-------------------------------------------
```

```
-- TRIGGER
--------------------------------------------
GO
CREATE TRIGGER Tenant_INSERT
ON Tenant
AFTER INSERT
AS
INSERT INTO History(Operation)
SELECT 'Добавлен арендатор '+ Full_Name_or_Organization + ' адрес ' + Address
FROM INSERTED
GO

--INSERT INTO Tenant(Tenant_ID, Full_Name_or_Organization, Phone, Address,
Client_Type_ID, District_ID) VALUES (9, 'Stefan Mikle', 783789, 'Moses Tunnel,
72', 2, 5);
--SELECT * FROM History;

GO
CREATE TRIGGER Tenant_Check
	ON Tenant AFTER INSERT
	AS
BEGIN
	DECLARE @full_name VARCHAR(100)
	SELECT @full_name=(SELECT Full_Name_or_Organization FROM
inserted)
	IF @full_name='Ivan Ivanov'
	BEGIN
		RAISERROR('Ivan is fined', 17, 1);
		ROLLBACK;
		RETURN
	END;
END
GO

GO
CREATE TRIGGER Renter_Address_UPDATE
ON Renter
AFTER UPDATE
AS
INSERT INTO History(Operation)
SELECT 'Изменен адрес арендодателя '+ Full_Name_or_Organization + ' новый
адрес ' + Address
FROM inserted
GO

--------------------------------------
--VIEW
```

```
--------------------------------------
GO
CREATE VIEW V_District_Dorset(Rental_Start_Date, Return_Date,
Full_Name_or_Organization, Phone, Price)
AS
 SELECT Car_Rental.Rental_Start_Date, Car_Rental.Return_Date,
Tenant.Full_Name_or_Organization, Tenant.Phone, Car.Price
  FROM Car_Rental JOIN Tenant ON Car_Rental.Tenant_ID = Tenant.Tenant_ID
                              JOIN District ON District.District_ID = Tenant.District_ID
                              JOIN Car ON Car_Rental.Car_ID = Car.Car_ID

                                    WHERE Name='Dorset';
GO
SELECT * FROM V_District_Dorset;

GO
CREATE VIEW V_Physical_Person
AS
 SELECT  Car_Rental.Rental_Start_Date, Car_Rental.Return_Date,
Renter.Full_Name_or_Organization, Renter.Phone
  FROM Car_Rental JOIN Renter ON Car_Rental.Renter_ID = Renter.Renter_ID
                      JOIN Client_Type ON Renter.Client_Type_ID =
Client_Type.Client_Type_ID
                              WHERE Name = 'physical person';
GO
SELECT * FROM V_Physical_Person;

GO
CREATE VIEW V_Order_List
AS
 SELECT Car_Rental.Rental_Start_Date, Car_Rental.Return_Date,
Renter.Full_Name_or_Organization as r_name, Renter.Phone as renter_phone,
Renter.Address as r_address,  Tenant.Full_Name_or_Organization as t_name,
Tenant.Phone as tenant_phone, Tenant.Address as t_address
  FROM Car_Rental, Renter, Tenant
   WHERE  Car_Rental.Renter_ID = Renter.Renter_ID AND Car_Rental.Tenant_ID
= Tenant.Tenant_ID;
GO

-------------------------------------------
-- UPDATE
-------------------------------------------
--(1)Добавляем префикс к адресу арендодателя с ID района равным 2 или 3
UPDATE Renter
SET Address = '(ID_2/3) ' + Address
WHERE District_ID = 2 OR District_ID = 3;
--(2)Добавляем префикс к регистрационному номеру машины, если пробег
```

```
автомобиля больше 100 000 км
UPDATE Car
SET Registration_Number = '(U) ' + Registration_Number
WHERE Condition_ID IN
(SELECT Condition_ID FROM Condition WHERE Mileage > 100000);

--(3)Поднимаем цену аренды автомобилей на 200 рублей, date_last_inspection
после 2020
UPDATE Car
SET Price = Price + 200
WHERE Condition_ID IN
(SELECT Condition_ID FROM Condition WHERE YEAR(Date_Last_Inspection)>
2020);
-------------------------------------------
--DELETE
-------------------------------------------

--(4)удаляем арендаторов, которые сделали заказ на аренду больше месяца
DELETE FROM Tenant WHERE Tenant_ID IN (SELECT Tenant.Tenant_ID FROM
Tenant, Car_Rental WHERE
Tenant.Tenant_ID = Car_Rental.Tenant_ID AND
(MONTH(Car_Rental.Return_Date) - MONTH(Car_Rental.Rental_Start_Date) > 1
OR YEAR(Car_Rental.Return_Date) - YEAR(Car_Rental.Rental_Start_Date) > 0));

-------------------------------------------
-- SELECT
-------------------------------------------
--(5)Максимальная и минимальная цена аренды автомобиля
SELECT MIN(Car.Price) as 'max_price',  MAX(Car.Price) as 'min_price'
FROM Car, Car_Rental, Tenant, Client_Type WHERE Car.Car_ID =
Car_Rental.Car_ID AND Car_Rental.Tenant_ID = Tenant.Tenant_ID AND
Tenant.Client_Type_ID = Client_Type.Client_Type_ID
 AND Client_Type.Name = 'physical person';

--(6)сгруппировать машины по количеству заказов
SELECT COUNT(*) as 'total', Brand.make, Brand.model
FROM Car_Rental JOIN Car ON Car_Rental.Car_ID = Car.Car_ID
                              JOIN Brand ON Car.Brand_ID = Brand.Brand_ID
       GROUP BY Brand.make, Brand.model
       HAVING COUNT(*) > 1;

--(7) Арендаторы, отсортированные по дате начала аренды
SELECT Rental_Start_Date, Return_Date, Full_Name_or_Organization, Phone,
Price FROM V_District_Dorset
ORDER BY Rental_Start_Date;

--(8) Получаем к какому типу клиента относятся арендодатели
```

```
SELECT Renter.Full_Name_or_Organization, Renter.Phone, Renter.Address,
Client_Type.Name
AS 'Client_Type' FROM Renter
FULL OUTER JOIN Client_Type
ON Renter.Client_Type_ID = Client_Type.Client_Type_ID;

--(9) Получаем процент  заказов из района Dorset
SELECT
		(SELECT CONVERT(decimal, COUNT(*)) FROM V_District_Dorset)/
		(SELECT COUNT(*) FROM Car_Rental) * 100 AS 'Dorset orders %';

--(10) Хотим найти всех арендаторов и арендодателей, которые делали заказы
весной
SELECT * FROM
		(SELECT Tenant.Full_Name_or_Organization, Tenant.Phone,
Tenant.Address FROM
		Tenant, Car_Rental WHERE Car_Rental.Tenant_ID = Tenant.Tenant_ID
AND
		MONTH(Car_Rental.Rental_Start_Date) > 2 AND
MONTH(Car_Rental.Return_Date) < 6
		UNION
		SELECT Renter.Full_Name_or_Organization, Renter.Phone,
Renter.Address FROM
		Renter, Car_Rental WHERE Car_Rental.Renter_ID = Renter.Renter_ID
AND
		MONTH(Car_Rental.Rental_Start_Date) > 2 AND
MONTH(Car_Rental.Return_Date) < 6)
		AS union_table;

--(11) Выбрать все машины, у которых дата последнего техосмотра была до
2020 года
SELECT Car.Registration_Number, Car.Brand_ID, Condition.Date_Last_Inspection
FROM
Car, Condition
WHERE Car.Condition_ID = Condition.Condition_ID AND Condition.Condition_ID
NOT IN
(SELECT Condition.Condition_ID FROM Condition WHERE
YEAR(Condition.Date_Last_Inspection) > 2020)

--(12) Имя арендатора и стоимость аренды за указанный период
SELECT Tenant.Full_Name_or_Organization,  Car.Price*(DATEDIFF(day,
Car_Rental.Rental_Start_Date, Car_Rental.Return_Date)) FROM
	Car_Rental, Car, Tenant
		WHERE Car_Rental.Car_ID = Car.Car_ID AND Tenant.Tenant_ID =
Car_Rental.Tenant_ID;

-------------------------------------------
```

```
-- DROP
--------------------------------------------
DROP PROCEDURE AddNewTenant;
DROP FUNCTION CalculateCost;
DROP FUNCTION CheckInspection;

DROP INDEX idx_brand_name;
DROP INDEX idx_car_registr_number;
DROP INDEX idx_car_rental_period;

DROP VIEW V_District_Dorset;
DROP VIEW V_Physical_Person;
DROP VIEW V_Order_List;

DROP TRIGGER Tenant_INSERT;
DROP TRIGGER Tenant_Check;
DROP TRIGGER Renter_Address_UPDATE;

DROP TABLE Car_Rental;
DROP TABLE Car;
DROP TABLE Tenant;
DROP TABLE Renter;
DROP TABLE Brand;
DROP TABLE Condition;
DROP TABLE Client_Type;
DROP TABLE District;
```