**\! chcp 1251**

create database car\_rental; /создание базы данных

\c car\_rental /переход в бд

create table users( /таблица клиенты

user\_id serial primary key,

u\_surname varchar(20),

u\_name varchar(20),

u\_patronymic varchar(20),

u\_adres varchar(255),

u\_phone varchar(12));

create table orders( /таблица заказ

order\_id serial primary key,

o\_date\_of\_issue timestamp,

o\_date\_return timestamp,

o\_cost integer,

o\_date\_of\_actual\_return timestamp,

o\_fine integer,

o\_discount integer);

create table status( /таблица статус

status\_id serial primary key,

s\_name varchar(255));

create table brand( /таблица марка

brand\_id serial primary key,

b\_name varchar(255));

create table auto( /таблица автомобиль

auto\_id serial primary key,

a\_name varchar(255),

a\_year integer,

a\_cost\_hour integer);

create table type\_of\_car( /таблица тип авто

type\_of\_car\_id serial primary key,

t\_name varchar(255));

alter table orders add column orders\_o\_user\_id\_fk int references users(user\_id);

alter table orders add column orders\_o\_status\_id\_fk int references status(status\_id);

alter table orders add column orders\_o\_car\_id\_fk int references auto(auto\_id);

alter table auto add column auto\_a\_brand\_id\_fk int references brand(brand\_id);

alter table auto add column auto\_a\_type\_of\_car\_id\_fk int references type\_of\_car(type\_of\_car\_id);

insert into users(u\_surname, u\_name, u\_patronymic, u\_adres, u\_phone) values('Попов ', 'Иван ', 'Иванович ', 'Москва, ул. Ленина, д. 5 ', 89124586219);

insert into users(u\_surname, u\_name, u\_patronymic, u\_adres, u\_phone) values('Логинов', 'Александр', 'Евгеньевич', 'Киров, ул. Ломоносова, д. 16', 89221453287);

insert into status(s\_name) values('Завершен'),('В прокате'),('Отменен');

insert into brand(b\_name) values('Audi'),('Toyota'),('BMW');

insert into type\_of\_car(t\_name) values('Седан'),('Внедорожник'),('Кроссовер');

insert into auto(a\_name, auto\_a\_brand\_id\_fk, auto\_a\_type\_of\_car\_id\_fk, a\_year, a\_cost\_hour) values('Audi Q5', 1, 2, 2023, 400);

insert into auto(a\_name, auto\_a\_brand\_id\_fk, auto\_a\_type\_of\_car\_id\_fk, a\_year, a\_cost\_hour) values('Toyota Mark 2', 2, 1, 1998, 200);

insert into auto(a\_name, auto\_a\_brand\_id\_fk, auto\_a\_type\_of\_car\_id\_fk, a\_year, a\_cost\_hour) values('BMW iX', 3, 3, 2023, 500);

insert into orders(o\_date\_of\_issue, o\_date\_return, o\_cost, o\_date\_of\_actual\_return, orders\_o\_user\_id\_fk, orders\_o\_status\_id\_fk, o\_fine, o\_discount, orders\_o\_car\_id\_fk) values('2024-05-06 12:00:00', '2024-05-10 12:00:00', 19200, null, 2, 2, null, 0, 2);

insert into orders(o\_date\_of\_issue, o\_date\_return, o\_cost, o\_date\_of\_actual\_return, orders\_o\_user\_id\_fk, orders\_o\_status\_id\_fk, o\_fine, o\_discount, orders\_o\_car\_id\_fk) values('2024-03-17 00:00:00', '2024-03-19 00:00:00', 9600, '2024-03-18 00:00:00', 1, 1, null, 0, 1);

select a\_name, a\_year from auto where a\_year > 2015;

select o\_cost,o\_date\_of\_issue from orders where extract(month from date\_trunc('month', o\_date\_of\_issue))= 5 and extract(year from date\_trunc('year', o\_date\_of\_issue)) = 2024;

select users.u\_surname, users.u\_name, orders.o\_date\_of\_issue from users inner join orders on users.user\_id = orders.orders\_o\_user\_id\_fk where orders.o\_cost > 5000;

select u\_surname, u\_name from users where user\_id in(select orders\_o\_user\_id\_fk from orders where orders\_o\_status\_id\_fk = 1);

CREATE OR REPLACE FUNCTION get\_user\_orders(p\_user\_id INT)

RETURNS TABLE(

order\_id INT,

date\_of\_issue TIMESTAMP,

date\_return TIMESTAMP,

cost INT,

date\_of\_actual\_return TIMESTAMP,

fine INT,

discount INT

) AS $$

BEGIN

RETURN QUERY

SELECT

orders.order\_id,

orders.o\_date\_of\_issue,

orders.o\_date\_return,

orders.o\_cost,

orders.o\_date\_of\_actual\_return,

orders.o\_fine,

orders.o\_discount

FROM

orders

WHERE

orders.orders\_o\_user\_id\_fk = p\_user\_id;

END; $$

LANGUAGE 'plpgsql';

select \* from get\_user\_orders(2);

CREATE OR REPLACE FUNCTION get\_cars\_by\_brand(p\_brand\_id INT)

RETURNS TABLE(

auto\_id INT,

a\_name VARCHAR(255),

a\_year INT,

a\_cost\_hour INT

) AS $$

BEGIN

RETURN QUERY

SELECT

auto.auto\_id,

auto.a\_name,

auto.a\_year,

auto.a\_cost\_hour

FROM

auto

WHERE

auto.auto\_a\_brand\_id\_fk = p\_brand\_id;

END; $$

LANGUAGE 'plpgsql';

select \* from get\_cars\_by\_brand (1);

CREATE OR REPLACE FUNCTION update\_return\_date()

RETURNS TRIGGER AS $$

BEGIN

IF NEW.orders\_o\_status\_id\_fk = (SELECT status\_id FROM status WHERE s\_name = 'Завершен') THEN

NEW.o\_date\_of\_actual\_return = NOW();

END IF;

RETURN NEW;

END;

$$ LANGUAGE plpgsql;

CREATE TRIGGER orders\_status\_update

BEFORE UPDATE ON orders

FOR EACH ROW

EXECUTE FUNCTION update\_return\_date();

UPDATE orders

SET orders\_o\_status\_id\_fk = (SELECT status\_id FROM status WHERE s\_name = 'Завершен')

WHERE order\_id = 1;

CREATE OR REPLACE FUNCTION update\_order\_cost()

RETURNS TRIGGER AS $$

DECLARE

hours\_diff INT;

car\_cost\_per\_hour INT;

BEGIN

SELECT a\_cost\_hour INTO car\_cost\_per\_hour FROM auto WHERE auto\_id = NEW.orders\_o\_car\_id\_fk;

hours\_diff = EXTRACT(EPOCH FROM (NEW.o\_date\_return - NEW.o\_date\_of\_issue))/3600;

NEW.o\_cost = hours\_diff \* car\_cost\_per\_hour;

RETURN NEW;

END;

$$ LANGUAGE plpgsql;

CREATE TRIGGER orders\_cost\_update

BEFORE INSERT ON orders

FOR EACH ROW

EXECUTE FUNCTION update\_order\_cost();

insert into orders(o\_date\_of\_issue, o\_date\_return, o\_date\_of\_actual\_return, orders\_o\_user\_id\_fk, orders\_o\_status\_id\_fk, o\_fine, o\_discount, orders\_o\_car\_id\_fk) values('2024-07-16 00:00:00', '2024-07-19 00:00:00', null, 1, 2, null, 0, 3);

select \* from orders;

CREATE USER new\_user WITH PASSWORD '456';

GRANT SELECT, UPDATE ON orders TO new\_user;