

Санкт-Петербургский государственный
университет
Математико-механический факультет

Базы данных и СУБД
Отчёт по самостоятельному заданию

Студент: Овсянников К. А., гр. 22Б-10 ММ

весна 2024 г.


```

);

/* Хот нул убрал */
CREATE TABLE IF NOT EXISTS vehicle (
    id bigint GENERATED ALWAYS AS IDENTITY NOT NULL UNIQUE,
    name text NOT NULL,
    model_fkey bigint NOT NULL,
    year bigint NOT NULL,
    price bigint NOT NULL,
    condition bigint NOT NULL,
    color text NOT NULL,
    dealer_fkey bigint,
    status_fkey bigint NOT NULL, /* исправить бул на инт на диаграмме*/
    PRIMARY KEY (id)
);

CREATE TABLE IF NOT EXISTS model (
    id bigint GENERATED ALWAYS AS IDENTITY NOT NULL UNIQUE,
    name text NOT NULL,
    displacement bigint NOT NULL,
    power bigint NOT NULL,
    brand_fkey bigint NOT NULL,
    type_fkey bigint NOT NULL,
    service_quotient double precision NOT NULL,
    PRIMARY KEY (id)
);

CREATE TABLE IF NOT EXISTS service (
    id bigint GENERATED ALWAYS AS IDENTITY NOT NULL UNIQUE,
    address text NOT NULL UNIQUE,
    /* добавить поле name */
    PRIMARY KEY (id)
);

CREATE TABLE IF NOT EXISTS mechanic (
    id bigint GENERATED ALWAYS AS IDENTITY NOT NULL UNIQUE,
    name text NOT NULL,
    service_fkey bigint NOT NULL,
    salary bigint NOT NULL,
    phone_number text NOT NULL,
    PRIMARY KEY (id)
);

CREATE TABLE IF NOT EXISTS operation (
    id bigint GENERATED ALWAYS AS IDENTITY NOT NULL UNIQUE,
    name text NOT NULL UNIQUE,
    base_price bigint NOT NULL,
    PRIMARY KEY (id)
);

```

```

CREATE TABLE IF NOT EXISTS serviceList (
    service_fkey bigint NOT NULL,
    operation_fkey bigint NOT NULL
);

CREATE TABLE IF NOT EXISTS client (
    id bigint GENERATED ALWAYS AS IDENTITY NOT NULL UNIQUE,
    name text NOT NULL,
    phone_number text NOT NULL,
    PRIMARY KEY (id)
);

CREATE TABLE IF NOT EXISTS repair (
    id bigint GENERATED ALWAYS AS IDENTITY NOT NULL UNIQUE,
    vehicle_fkey bigint NOT NULL,
    client_fkey bigint NOT NULL,
    mechanic_fkey bigint NOT NULL,
    operation_fkey bigint NOT NULL,
    date date NOT NULL,
    PRIMARY KEY (id)
);

CREATE TABLE IF NOT EXISTS deal (
    id bigint GENERATED ALWAYS AS IDENTITY NOT NULL UNIQUE,
    vehicle_fkey bigint NOT NULL,
    client_fkey bigint NOT NULL,
    date date NOT NULL,
    PRIMARY KEY (id)
);

CREATE TABLE IF NOT EXISTS status (
    id bigint GENERATED ALWAYS AS IDENTITY NOT NULL UNIQUE,
    status text NOT NULL UNIQUE,
    PRIMARY KEY (id)
);

CREATE TABLE IF NOT EXISTS type (
    id bigint GENERATED ALWAYS AS IDENTITY NOT NULL UNIQUE,
    type text NOT NULL UNIQUE,
    PRIMARY KEY (id)
);

CREATE TABLE IF NOT EXISTS brand (
    id bigint GENERATED ALWAYS AS IDENTITY NOT NULL UNIQUE,
    name text NOT NULL,
    country text NOT NULL,
    description text,
    PRIMARY KEY (id)
);

```

```

CREATE TABLE IF NOT EXISTS type_skills (
    mechanic_fkey bigint NOT NULL,
    type_fkey bigint NOT NULL
);

CREATE TABLE IF NOT EXISTS brand_skills (
    mechanic_fkey bigint NOT NULL,
    brand_fkey bigint NOT NULL
);

ALTER TABLE type_skills ADD CONSTRAINT fkey_mechanictypeskills_mechanic FOREIGN KEY (mech
ALTER TABLE type_skills ADD CONSTRAINT fkey_mechanictypeskills_type FOREIGN KEY (type_fk

ALTER TABLE brand_skills ADD CONSTRAINT fkey_mechanicbrandskills_mechanic FOREIGN KEY (m
ALTER TABLE brand_skills ADD CONSTRAINT fkey_mechanicbrandskills_brand FOREIGN KEY (bran

ALTER TABLE vehicle ADD CONSTRAINT fkey_vehicle_model FOREIGN KEY (model_fkey) REFERENCE
ALTER TABLE vehicle ADD CONSTRAINT fkey_vehicle_dealer FOREIGN KEY (dealer_fkey) REFEREN
ALTER TABLE vehicle ADD CONSTRAINT fkey_vehicle_status FOREIGN KEY (status_fkey) REFEREN

ALTER TABLE model ADD CONSTRAINT fkey_model_brand FOREIGN KEY (brand_fkey) REFERENCES br
ALTER TABLE model ADD CONSTRAINT fkey_model_type FOREIGN KEY (type_fkey) REFERENCES type

ALTER TABLE mechanic ADD CONSTRAINT fkey_mechanic_service FOREIGN KEY (service_fkey) REF

ALTER TABLE serviceList ADD CONSTRAINT fkey_servicelist_service FOREIGN KEY (service_fke
ALTER TABLE serviceList ADD CONSTRAINT fkey_servicelist_operation FOREIGN KEY (operation

ALTER TABLE repair ADD CONSTRAINT fkey_repair_vehicle FOREIGN KEY (vehicle_fkey) REFEREN
ALTER TABLE repair ADD CONSTRAINT fkey_repair_mechanic FOREIGN KEY (mechanic_fkey) REFER
ALTER TABLE repair ADD CONSTRAINT fkey_repair_client FOREIGN KEY (client_fkey) REFERENCE
ALTER TABLE repair ADD CONSTRAINT fkey_repair_operation FOREIGN KEY (operation_fkey) REF

ALTER TABLE deal ADD CONSTRAINT fkey_deal_vehicle FOREIGN KEY (vehicle_fkey) REFERENCES
ALTER TABLE deal ADD CONSTRAINT fkey_deal_client FOREIGN KEY (client_fkey) REFERENCES cl

```

2.2 Скрипт заполнения базы данных

```

INSERT INTO "VehicleType" (type) VALUES
    ("motorcycle"), ("car"), ("scooter"), ("trailer");

INSERT INTO "Brand" (name, country, description) VALUES
    ("Honda", "Japan", "nice reliable car and bikes manufacturer"),
    ("BA3", "Russia", "отечественный производитель нестареющей классики и других авт
    ("Hyundai", "South Korea"),
    ("Kawasaki", "Japan", "japanese street- and sportbikes manufacturer");

INSERT INTO "Service" (address) VALUES
    ("Москва"),

```

```
("Санкт-Петербург"),  
("Севастополь");
```

```
INSERT INTO "Dealer" (address) VALUES  
("Москва"),  
("Санкт-Петербург"),  
("Краснодар");
```

```
INSERT INTO "Operation" (name, base_price) VALUES  
("Замена масла в двигателе", 2000),  
("Замена резины на одном колесе", 1500),  
("Регулировка зазоров клапанов", 2500),  
("Раскоксовка двигателя", 3500);
```

```
INSERT INTO "Status" (status) VALUES  
("in stock"),  
("sold"),  
("unknown");
```

```
INSERT INTO "Client" (name, phone_number) VALUES  
("Константин", "89113030888"),  
("Александр", "89111234567"),  
("Станислав", "89247654321");
```

```
INSERT INTO "Mechanic" (name, service_key, salary, phone_number) VALUES  
("Василий", 2, 50000, "89119876543"),  
("Андрей", 2, 80000, "89113456789"),  
("Даниил", 1, 75000, "89771234567"),  
("Егор", 3, 50000, "88692345678");
```

```
INSERT INTO "MechanicTypeSkills" (mechanic_fkey, type_fkey) VALUES  
(1, 1),  
(1, 2),  
(1, 3),  
(2, 1),  
(3, 1),  
(3, 2),  
(4, 1),  
(4, 3);
```

```
INSERT INTO "MechanicBrandSkills" (mechanic_fkey, brand_fkey) VALUES  
(1, 1),  
(1, 2),  
(2, 1),  
(2, 2),  
(2, 3),  
(3, 1),  
(3, 2),  
(3, 3),  
(3, 4),
```

```
(4, 1),  
(4, 4);
```

```
INSERT INTO "Model" (name, year, displacement, hp, brand_fkey, type_fkey, service_quotie  
("Honda CB400SF", 400, 53, 1, 1, 0.5),  
("BA3 2101", 1200, 64, 2, 2, 1),  
("Hyundai Solaris", 1600, 122, 3, 2, 1.2);
```

```
INSERT INTO "Vehicle" (name, model_fkey, year, price, condition, color, dealer_fkey, sta  
("Honda CB400SF Hyper VTEC Spec 2", 1, 2002, 340000, 7, "black", 2, 2),  
("BA3 2101 Жигули", 2, 1977, 400000, 6, "blue", 2, 1);  
("Hyundai Solaris", 3, 2020, 1500000, 8, "white", NULL, 3),  
("");
```

2.3 Скрипт удаления всех таблиц в базе данных

```
DROP TABLE deal;  
DROP TABLE repair;  
DROP TABLE vehicle;  
DROP TABLE status;  
DROP TABLE dealer;  
DROP TABLE model;  
DROP TABLE type_skills;  
DROP TABLE brand_skills;  
DROP TABLE type;  
DROP TABLE brand;  
DROP TABLE serviceList;  
DROP TABLE operation;  
DROP TABLE mechanic;  
DROP TABLE client;  
DROP TABLE service;
```

2.4 Скрипт очистки всех таблиц в базе данных

```
TRUNCATE TABLE deal;  
TRUNCATE TABLE repair;  
TRUNCATE TABLE vehicle;  
TRUNCATE TABLE status;  
TRUNCATE TABLE dealer;  
TRUNCATE TABLE model;  
TRUNCATE TABLE type_skills;  
TRUNCATE TABLE brand_skills;  
TRUNCATE TABLE type;  
TRUNCATE TABLE brand;  
TRUNCATE TABLE serviceList;  
TRUNCATE TABLE operation;  
TRUNCATE TABLE mechanic;  
TRUNCATE TABLE client;  
TRUNCATE TABLE service;
```

2.5 Пользовательские запросы

2.6 Процедуры

2.6.1 Статистические функции

/ Скрипты подсчёта статистических величин */*

```
CREATE OR REPLACE FUNCTION dealer_revenue(dealer_id int, date_part int, mode text) RETURN
```

```
AS $$
```

```
BEGIN
```

```
IF mode != 'month' AND mode != 'year' THEN
```

```
    RAISE EXCEPTION 'Incorrect date mode!';
```

```
END IF;
```

```
RETURN (select sum(price) FROM (deal JOIN vehicle ON deal.vehicle_fkey = vehicle.id) WHERE
```

```
END
```

```
$$ LANGUAGE plpgsql;
```

```
CREATE OR REPLACE FUNCTION yearly_dealer_revenue(dealer_id int, year int) RETURNS int
```

```
AS $$
```

```
BEGIN
```

```
RETURN dealer_revenue(dealer_id, year, 'year');
```

```
END
```

```
$$ LANGUAGE plpgsql;
```

```
CREATE OR REPLACE FUNCTION monthly_dealer_revenue(dealer_id int, month int) RETURNS int
```

```
AS $$
```

```
BEGIN
```

```
RETURN dealer_revenue(dealer_id, month, 'month');
```

```
END
```

```
$$ LANGUAGE plpgsql;
```

```
CREATE OR REPLACE FUNCTION service_revenue(service_id int, date_part int, mode text) RET
```

```
AS $$
```

```
BEGIN
```

```
IF mode != 'month' AND mode != 'year' THEN
```

```
    RAISE EXCEPTION 'Incorrect date mode!';
```

```
END IF;
```

/ временная таблица, хранящая сервисные коэффициенты для каждого ТС */*

```
CREATE TEMP TABLE vehicle_quotient AS SELECT vehicle_fkey, service_quotient FROM
```

```
((repair JOIN vehicle ON vehicle.id = vehicle_fkey) JOIN model ON model.id = model_fkey)
```

/ таблица для получения сервиса, в котором производилось обслуживание */*

```
CREATE TEMP TABLE vehicle_service AS SELECT mechanic_fkey, service_fkey FROM
```

```
(repair JOIN mechanic ON mechanic.id = mechanic_fkey);
```

```
RETURN (select sum(operation.base_price * vehicle_quotient.service_quotient) FROM
```

```
((repair JOIN vehicle_quotient ON repair.vehicle_fkey = vehicle_quotient.vehicle_fkey)
```

```
ON repair.mechanic_fkey = vehicle_service.mechanic_fkey) JOIN operation ON operation.id
```

```
WHERE service_fkey = service_id AND DATE_PART(mode, date::date) = date_part);
```



```
END
```

```
$$ LANGUAGE plpgsql;
```

```
CREATE OR REPLACE FUNCTION yearly_service_revenue(service_id int, year int) RETURNS int
```

```
AS $$
```

```
BEGIN
```

```
RETURN service_revenue(service_id, year, 'year');
```

```
END
```

```
$$ LANGUAGE plpgsql;
```

```
CREATE OR REPLACE FUNCTION monthly_service_revenue(service_id int, month int) RETURNS int
```

```
AS $$
```

```
BEGIN
```

```
RETURN service_revenue(service_id, month, 'month');
```

```
END
```

```
$$ LANGUAGE plpgsql;
```

2.7 Триггеры

```
/* Триггеры базы данных */
```

```
/* Изменение статуса автомобиля при совершении на него сделки */
```

```
/* Ограничение на совершение сделок с автомобилями, которых нет в наличии*/
```

2.8 Представления