Санкт-Петербургский государственный университет
Математико-механический факультет

Базы данных и СУБД
Отчёт по самостоятельному заданию

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

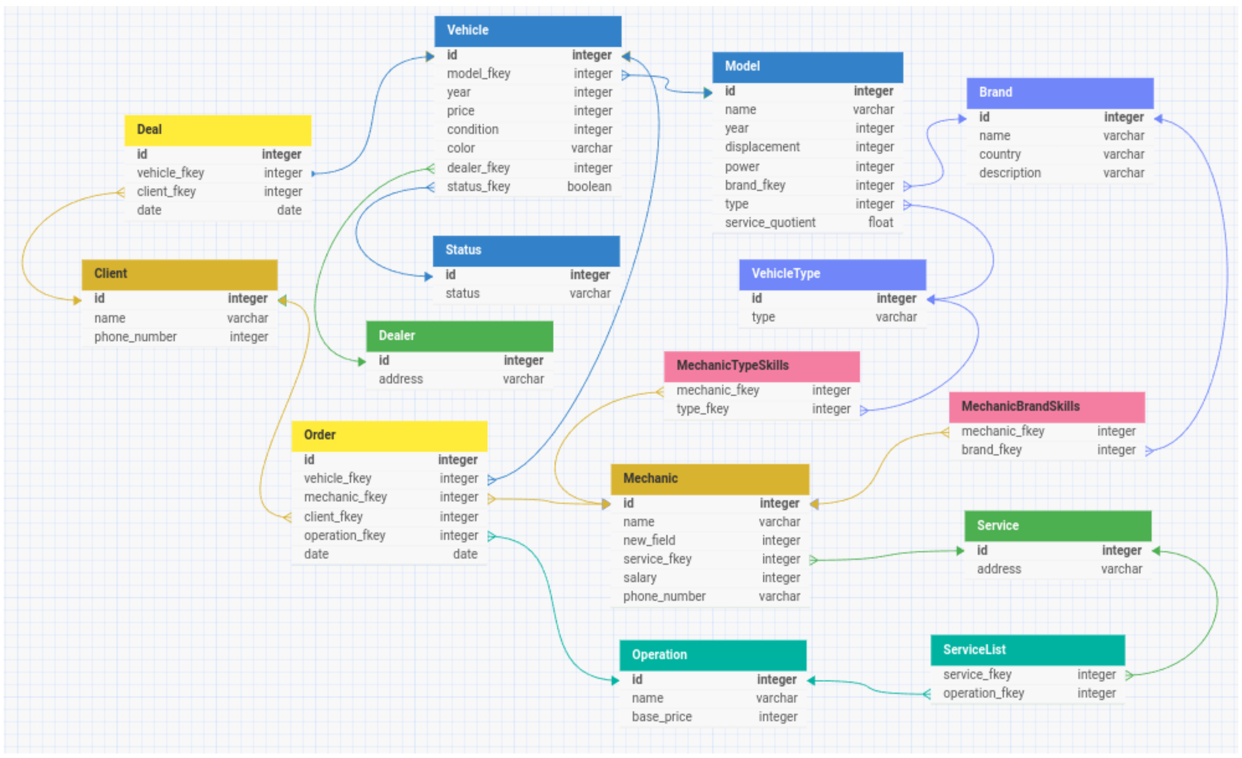
весна 2024 г.

1 Описание предметной области

1.1 Требования заказчика

Компания-заказчик занимается продажей и ремонтом авто- и мототехники. Реализация осуществляется в отделениях, расположенных по всей стране. Также компания имеет сеть сервисов, которые занимаются ремонтом и обслуживанием техники. В каждом из сервисов работают опытные специалисты, осуществляющие ремонт техники определённых типов и марок.
Расчёт стоимости за услугу осуществляется из базовой цены услуги и умножающего коэффициента соответствующего конкретной модели техники.
Требуется создать базу данных для учёта имеющихся в наличии ТС, контроля зарплат сотрудников и отображения набора услуг.

1.2 ER-диаграмма



2 Скрипты

2.1 Скрипт создания таблиц и ограничений в базе данных

```
/* Создание таблиц */
CREATE TABLE IF NOT EXISTS dealer (
        id bigint GENERATED ALWAYS AS IDENTITY NOT NULL UNIQUE,
        address text NOT NULL UNIQUE,
        PRIMARY KEY (id)
);

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,
        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,
        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 service_list (
        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
);


/* Создание ограничений на поля */
ALTER TABLE type_skills ADD CONSTRAINT fkey_mechanictypeskills_mechanic FOREIGN KEY (mec
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 vehicle ADD CONSTRAINT check_condition_range CHECK(condition > 0 AND CONDITI

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 service_list ADD CONSTRAINT fkey_servicelist_service FOREIGN KEY (service_fk
ALTER TABLE service_list ADD CONSTRAINT fkey_servicelist_operation FOREIGN KEY (operatio

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

/* Создание индексов к наиболее часто используемым таблицам (deal, repair, vehicle) */
CREATE INDEX deal_idx ON deal (id, client_fkey);
CREATE INDEX repair_idx ON repair (id, client_fkey);
CREATE INDEX vehicle_idx ON vehicle (id, dealer_fkey, price);
```

2.2 Скрипт заполнения базы данных

```
INSERT INTO type (type) VALUES
        ('motorcycle'), ('car'), ('scooter'), ('trailer');

INSERT INTO brand (name, country, description) VALUES
        ('Honda', 'Japan', 'nice reliable car and bikes manufacturer'),
        ('ВАЗ', 'Russia', 'отечественный производитель нестареющей классики и других авт
        ('Hyundai', 'South Korea', 'korean car manufacturer'),
        ('Kawasaki', 'Japan', 'japanese street- and sportbikes manufacturer'),
        ('Dodge', 'US', 'american car manufacturer'),
        ('Toyota', 'Japan', 'japanese reliable car manufacturer');
```

```sql
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),
        ('Замена колёс', 5000);

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_fkey, salary, phone_number) VALUES
        ('Василий', 2, 50000, '89119876543'),
        ('Андрей', 2, 80000, '89113456789'),
        ('Даниил', 1, 75000, '89771234567'),
        ('Егор', 3, 50000, '88692345678');

INSERT INTO type_skills (mechanic_fkey, type_fkey) VALUES
        (1, 1),
        (1, 2),
        (1, 3),
        (2, 1),
        (2, 3),
        (3, 1),
        (3, 2),
        (4, 1),
        (4, 3);

INSERT INTO brand_skills (mechanic_fkey, brand_fkey) VALUES
        (1, 1),
        (1, 2),
        (2, 1),
        (2, 2),
        (2, 3),
        (3, 1),
        (3, 2),
        (3, 3),
        (3, 4),
        (3, 5),
        (3, 6),
        (4, 1),
        (4, 4);

INSERT INTO service_list (service_fkey, operation_fkey) VALUES
        (1, 1),
        (1, 2),
        (1, 3),
        (1, 4),
        (1, 5),
        (2, 1),
        (2, 2),
        (2, 3),
        (3, 1),
        (3, 2);


INSERT INTO model (name, displacement, power, brand_fkey, type_fkey, service_quotient) V
        ('Honda CB400SF', 400, 53, 1, 1, 0.5),
        ('ВАЗ 2101', 1200, 64, 2, 2, 1),
        ('Dodge Ram', 5700, 400, 5, 2, 4),
        ('Hyundai Solaris', 1600, 122, 3, 2, 1.2),
        ('Toyota Mark 2', 3500, 280, 6, 2, 1.5),
        ('Лада Веста', 1600, 123, 2, 2, 1);

INSERT INTO vehicle (name, model_fkey, year, price, condition, color, dealer_fkey, statu
        ('Honda CB400SF Hyper VTEC Spec 2', 1, 2002, 340000, 7, 'black', 2, 2),
        ('ВАЗ 2101 Жигули', 2, 1977, 400000, 6, 'blue', 2, 2),
        ('Dodge Ram 1500 hemi', 3, 2024, 5000000, 7, 'black', 1, 2),
        ('Hyundai Solaris', 4, 2020, 1500000, 8, 'white', NULL, 3),
        ('Toyota Mark 2', 5, 1995, 600000, 6, 'white', 3, 1),
        ('Hyundai Solaris', 4, 2015, 550000, 6, 'red', 2, 1),
        ('Лада Веста', 6, 2019, 700000, 8, 'blue', 1, 1);

INSERT INTO deal (vehicle_fkey, client_fkey, date) VALUES
        (1, 1, '2024-03-04'),
        (2, 2, '2023-06-21'),
        (3, 3, '2023-09-01');

INSERT INTO repair (vehicle_fkey, client_fkey, mechanic_fkey, operation_fkey, date) VALU
        (1, 1, 1, 1, '2024-03-08'),
        (2, 2, 1, 3, '2023-12-26'),
        (1, 1, 2, 3, '2024-05-11'),
        (3, 3, 3, 5, '2023-12-01');
```

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 service_list;
DROP TABLE operation;
DROP TABLE mechanic;
DROP TABLE client;
DROP TABLE service;
```

2.4 Скрипт очистки всех таблиц в базе данных

```
TRUNCATE TABLE deal RESTART IDENTITY CASCADE;
TRUNCATE TABLE repair RESTART IDENTITY CASCADE;
TRUNCATE TABLE vehicle RESTART IDENTITY CASCADE;
TRUNCATE TABLE status RESTART IDENTITY CASCADE;
TRUNCATE TABLE dealer RESTART IDENTITY CASCADE;
TRUNCATE TABLE model RESTART IDENTITY CASCADE;
TRUNCATE TABLE type_skills RESTART IDENTITY CASCADE;
TRUNCATE TABLE brand_skills RESTART IDENTITY CASCADE;
TRUNCATE TABLE type RESTART IDENTITY CASCADE;
TRUNCATE TABLE brand RESTART IDENTITY CASCADE;
TRUNCATE TABLE service_list RESTART IDENTITY CASCADE;
TRUNCATE TABLE operation RESTART IDENTITY CASCADE;
TRUNCATE TABLE mechanic RESTART IDENTITY CASCADE;
TRUNCATE TABLE client RESTART IDENTITY CASCADE;
TRUNCATE TABLE service RESTART IDENTITY CASCADE;
```

2.5 Пользовательские запросы

```
/* Пользовательские запросы к базе данных */

/* Подсчёт количества механиков, работающих в каждом городе */
SELECT address, count(*) FROM service JOIN mechanic ON service.id = mechanic.service_fke

/* В каких городах смогут провести работы по замене масла в двигателе мотоцикла HONda */
SELECT address FROM service
JOIN mechanic ON service.id = mechanic.service_fkey
JOIN type_skills ON mechanic.id = type_skills.mechanic_fkey
JOIN type ON type.id = type_skills.type_fkey
JOIN brand_skills ON mechanic.id = brand_skills.mechanic_fkey
JOIN brand ON brand.id = brand_skills.brand_fkey
JOIN servicelist ON servicelist.service_fkey = service.id
JOIN operatiON ON operatiON.id = servicelist.operatiON_fkey
WHERE brand.name = 'HONda' AND type.type = 'motorcycle' AND operation.name = 'Замена мас

/* Вычислить сумму, потраченную на ремонт каждым из клиентов */
SELECT name, sum(price) FROM
(SELECT client.name, operatiON.base_price * model.service_quotient AS price FROM client
JOIN repair ON repair.client_fkey = client.id
JOIN vehicle ON vehicle.id = repair.vehicle_fkey
JOIN model ON model.id = vehicle.model_fkey
JOIN operatiON ON operatiON.id = repair.operatiON_fkey)
GROUP BY name;

/* Вычислить сумму, потраченную каждым из клиентов на покупку ТС */
SELECT client.name, sum(price) FROM client
JOIN deal ON client.id = deal.client_fkey
JOIN vehicle ON vehicle.id = deal.vehicle_fkey
GROUP BY client.name;

/* Отобразить суммарную стоимость всех ТС из каждой категории в порядке возрастания стои
SELECT type, sum(price) FROM vehicle
JOIN model ON model.id = vehicle.model_fkey
JOIN type ON type.id = model.type_fkey
GROUP BY type;

/* Суммарная стоимость услуг, оказанная каждым из механиков */
SELECT mechanic.name, sum(repair_price(repair.id::INT)) FROM mechanic
JOIN repair ON mechanic.id = repair.mechanic_fkey
JOIN operatiON ON operatiON.id = repair.operatiON_fkey
GROUP BY mechanic.name;

/* Отобразить услуги, доступные в сервисе в Санкт-Петербурге */
SELECT name FROM service
JOIN servicelist ON service.id = servicelist.service_fkey
JOIN operatiON ON operatiON.id = servicelist.operatiON_fkey
WHERE address = 'Санкт-Петербург';

/* Отобразить количество общее количество услуг и сделок каждого клиента */
SELECT name, count(name) FROM client
JOIN (SELECT client_fkey FROM deal UNION ALL (SELECT client_fkey FROM repair))
ON client.id = client_fkey GROUP BY name;

/* Отобразить навыки механиков по маркам ТС в сервисе Санкт-Петербурга  */
SELECT brand.name FROM
(SELECT * FROM service WHERE address = 'Санкт-Петербург') as service
JOIN mechanic ON service.id = mechanic.service_fkey
JOIN brand_skills ON brand_skills.mechanic_fkey = mechanic.id
JOIN brand ON brand.id = brand_skills.brand_fkey
GROUP BY brand.name;

/*  Отобразить все ТС дороже 350000 рублей, отсортировав их по возрастанию цены */
```

```
SELECT * FROM vehicle WHERE price > 350000 ORDER BY price;

/* Отобразить самое дорогое ТС */
SELECT vehicle.name FROM vehicle WHERE price >= ALL(select price FROM vehicle)

/*  Отобразить российсские машины новее 2000 года (с использованием т.-мн. операций)*/
SELECT * FROM
(SELECT * FROM vehicle JOIN model ON model.id = vehicle.model_fkey JOIN brand ON brand.i
EXCEPT
(SELECT * FROM vehicle JOIN model ON model.id = vehicle.model_fkey JOIN brand ON brand.i
```

2.6 Процедуры и функции

2.6.1 Процедуры добавления сущностей

```
/* Скрипты добавления основных сущностей */
CREATE OR REPLACE PROCEDURE add_client(name text, phone_number text)
LANGUAGE SQL AS
$$
INSERT INTO client (name, phone_number) VALUES (add_client.name, add_client.phone_number
$$;


CREATE OR REPLACE PROCEDURE
add_vehicle(name text, model int, year int, price int, condition int, color text, dealer
LANGUAGE SQL AS
$$
INSERT INTO vehicle (name, model_fkey, year, price, condition, color, dealer_fkey, statu
(add_vehicle.name, model, add_vehicle.year, add_vehicle.price,
add_vehicle.condition, add_vehicle.color, dealer, status)
$$;

CREATE OR REPLACE PROCEDURE add_model(name text, displacement int, power int, brand int,
LANGUAGE SQL AS
$$
INSERT INTO model (name, displacement, power, brand_fkey, type_fkey, service_quotient) V
(add_model.name, add_model.displacement, add_model.power, brand, type, quotient)
$$;

CREATE OR REPLACE PROCEDURE add_deal(vehicle int, client int, date date)
LANGUAGE SQL AS
$$
INSERT INTO deal (vehicle_fkey, client_fkey, date) VALUES
(vehicle, client, add_deal.date)
$$;

CREATE OR REPLACE PROCEDURE add_repair(vehicle int, client int, mechanic int, operation
LANGUAGE SQL AS
$$
```

```
INSERT INTO repair (vehicle_fkey, client_fkey, mechanic_fkey, operation_fkey, date) VALU
(vehicle, client, mechanic, operation, date)
$$;
```

2.6.2 Статистические функции

```
/* Скрипты подсчёта статистических величин */
CREATE OR REPLACE FUNCTION dealer_revenue(dealer_id int, date_part int, mode text) RETUR

AS $$
BEGIN
IF mode != 'month' AND mode != 'year' THEN
        RAISE EXCEPTION 'Incorrect date mode!';
END IF;
RETURN (select sum(price) FROM (deal JOIN vehicle ON deal.vehicle_fkey = vehicle.id) WHE
END
$$ 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 in
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 (операция добавления в dea
CREATE OR REPLACE FUNCTION change_status() RETURNS trigger
AS $$
DECLARE status TEXT;
BEGIN
SELECT status.status INTO status FROM (SELECT * FROM vehicle WHERE vehicle.id = NEW.vehi
IF status <> 'in stock' THEN
        RAISE EXCEPTION 'Vehicle is not in stock';
END IF;
UPDATE vehicle SET status_fkey = 2 WHERE id = NEW.vehicle_fkey;
RETURN NEW;
END
$$ LANGUAGE plpgsql;

CREATE OR REPLACE TRIGGER change_status_tr BEFORE INSERT ON deal
FOR EACH ROW EXECUTE PROCEDURE change_status();

/* Проверка корректности даты относительно сегодняшней */
CREATE OR REPLACE FUNCTION check_date() RETURNS trigger
AS $$
DECLARE status TEXT;
BEGIN
IF NEW.date > NOW() THEN
        RAISE EXCEPTION 'Incorrect date';
END IF;
```

```
RETURN NEW;
END
$$ LANGUAGE plpgsql;

CREATE OR REPLACE TRIGGER check_date_deal_tr BEFORE INSERT ON deal
FOR EACH ROW EXECUTE PROCEDURE check_date();

CREATE OR REPLACE TRIGGER check_date_repair_tr BEFORE INSERT ON repair
FOR EACH ROW EXECUTE PROCEDURE check_date();
```

2.8 Представления

```
CREATE OR REPLACE VIEW motorcycle_v AS SELECT vehicle.id, vehicle.name, model_fkey, year
CREATE OR REPLACE VIEW car_v AS SELECT vehicle.id, vehicle.name, model_fkey, year, price
CREATE OR REPLACE VIEW scooter_v AS SELECT vehicle.id, vehicle.name, model_fkey, year, p
```