МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»  
(ФГБОУ ВО «ВГТУ», ВГТУ)

Факультет информационных технологий и компьютерной безопасности

Кафедра компьютерных интеллектуальных технологий проектирования

ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №8

По дисциплине: «Системы хранения и обработки данных»

Тема: «Разработка физической структуры базы данных»

Выполнил работу студент группы мИИВТ-241: Боченков А.С.

подпись, дата

Принял: Короленко В.В.

подпись, дата

Воронеж 2024

Учебная задача

3. Сформировать SQL-запросы для создания таблиц в СУБД Postgres, соответствующих логической структуре базы данных, разработанной в предыдущей лабораторной работе.

CREATE TABLE cars (

id serial PRIMARY KEY,

brand text NOT NULL,

model text NOT NULL,

year\_create integer NOT NULL,

number\_gov varchar(9) NOT NULL

);

Данный SQL-запрос создает таблицу cars. Атрибут id это первичный ключ, служит в качестве уникального идентификатора. Условие NOT NULL означает, что значение данного атрибута не может быть NULL. Для атрибута number\_gov (государственный номер автомобиля) выбран тип varchar(9) так как, номер не может занимать больше 9 символов, ограничение служит для экономии места.

CREATE TABLE violations (

    id serial PRIMARY KEY,

    name text NOT NULL,

    payment numeric(10,2) NOT NULL,

    CONSTRAINT positive\_payment CHECK(payment > 0)

);

Для атрибута payment (стоимость штрафа) выбран тип numeric(10,2) который показывает, что максимальное число состоит из 10 цифр, где первые 8 до запятой и 2 после. Такой тип данных выбран для экономии места, так как не бывает суммы единичного штрафа, превышающий его. Команда CONSTRAINT positive\_payment CHECK(payment > 0) создает новое ограничение для атрибута payment с названием positive\_payment и следит за тем, чтобы значение было всегда больше 0.

CREATE TABLE insurance (

insurance\_num integer PRIMARY KEY,

body\_payment numeric(10,2) NOT NULL,

percent\_insurance numeric(5,2) NOT NULL DEFAULT 10,

payment numeric(10,2) GENERATED ALWAYS AS (body\_payment \* percent\_insurance / 100) STORED,

CONSTRAINT valid\_insurance CHECK(insurance\_num > 100000),

CONSTRAINT positive\_body\_payment CHECK(body\_payment > 0),

CONSTRAINT valid\_percent\_insurance CHECK(percent\_insurance > 0 AND percent\_insurance < 100)

);

Атрибут insurance\_num называется естественным первичным ключом, потому что этот номер гарантирует уникальность. Условие DEFAULT 10 означает, что если значение данному атрибуту не было задано, то будет выставлено 10. Для того, чтобы атрибут payment (страховой платеж) рассчитывался автоматически в зависимости от значений в атрибутах body\_payment и percent\_insurance мы используем условие GENERATED ALWAYS AS (значение). Ключевое слово STORED, определяющее тип хранения генерируемого столбца, является обязательным.

Условие CONSTRAINT valid\_insurance CHECK(insurance\_num > 100000) нужно для того, чтобы значение страховки было больше шестизначного числа.

CREATE TABLE humans (

id serial PRIMARY KEY,

fio text NOT NULL,

sex char(1) NOT NULL,

birthday date NOT NULL,

CONSTRAINT valid\_sex CHECK (sex IN ('m', 'w') )

);

CONSTRAINT valid\_sex CHECK (sex IN ('m', 'w') ) команда создает условие с названием valid\_sex, которое проверяет, чтобы поступающее значение было только m или w. Иначе выдает ошибку с названием условия.

CREATE TABLE facts (

id serial PRIMARY KEY,

car\_id integer REFERENCES cars (id),

driver integer REFERENCES humans (id),

owner integer REFERENCES humans (id),

date timestamp NOT NULL,

violation\_id integer REFERENCES violations (id),

drive\_right varchar(10) GENERATED ALWAYS AS

(CASE

WHEN owner = driver THEN 'Owner'

ELSE 'Attorney'

END)

STORED,

insurance\_num integer REFERENCES insurance (insurance\_num),

CONSTRAINT valid\_drive\_right CHECK

(drive\_right IN ('Owner', 'Attorney')

)

);

car\_id integer REFERENCES cars (id) – создает атрибут car\_id таблицы facts, который является внешним (ссылающимся) ключом и ссылается на значение id в таблице cars.

drive\_right varchar(10) – выбран такой тип данных, потому что значение атрибута не будут превышать 10 символов.

GENERATED ALWAYS AS создает условия расчета атрибута в зависимости от значений других атрибутов, в нашем случае owner и driver.

CASE

WHEN

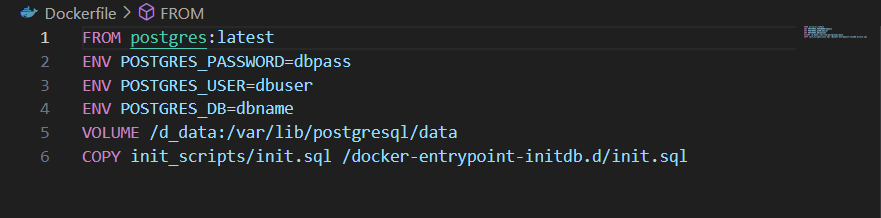
ELSE

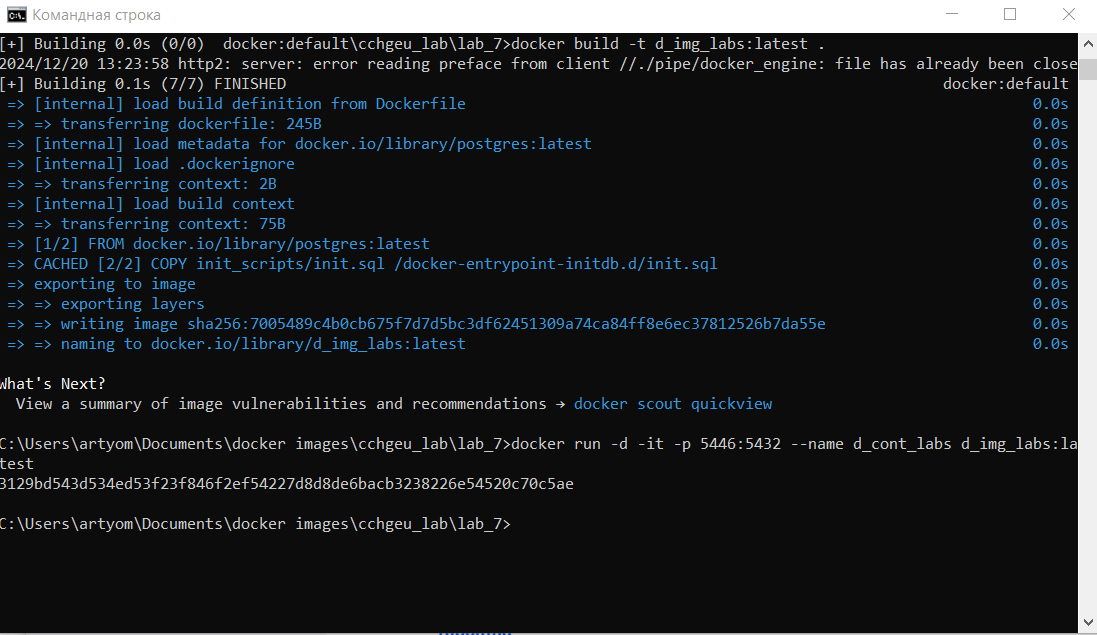
END это структура логического условия. В данном случае если значение атрибута owner равно значению атрибута driver, то присвоить значение Owner, иначе Attorney.

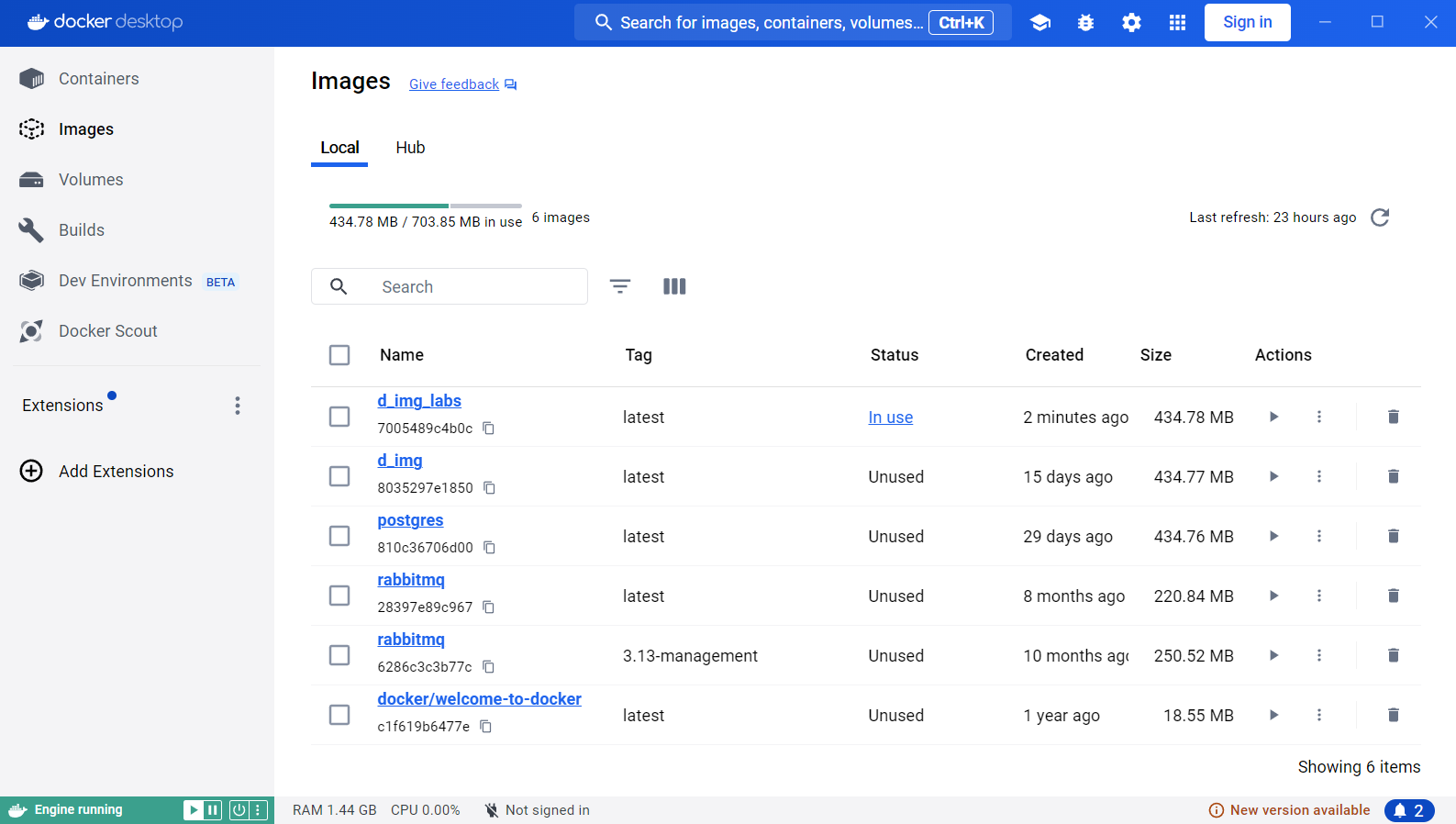
CONSTRAINT valid\_drive\_right CHECK (drive\_right IN ('Owner', 'Attorney') ) условие valid\_drive\_right проверяет, чтобы значение атрибута было Owner либо Attorney.

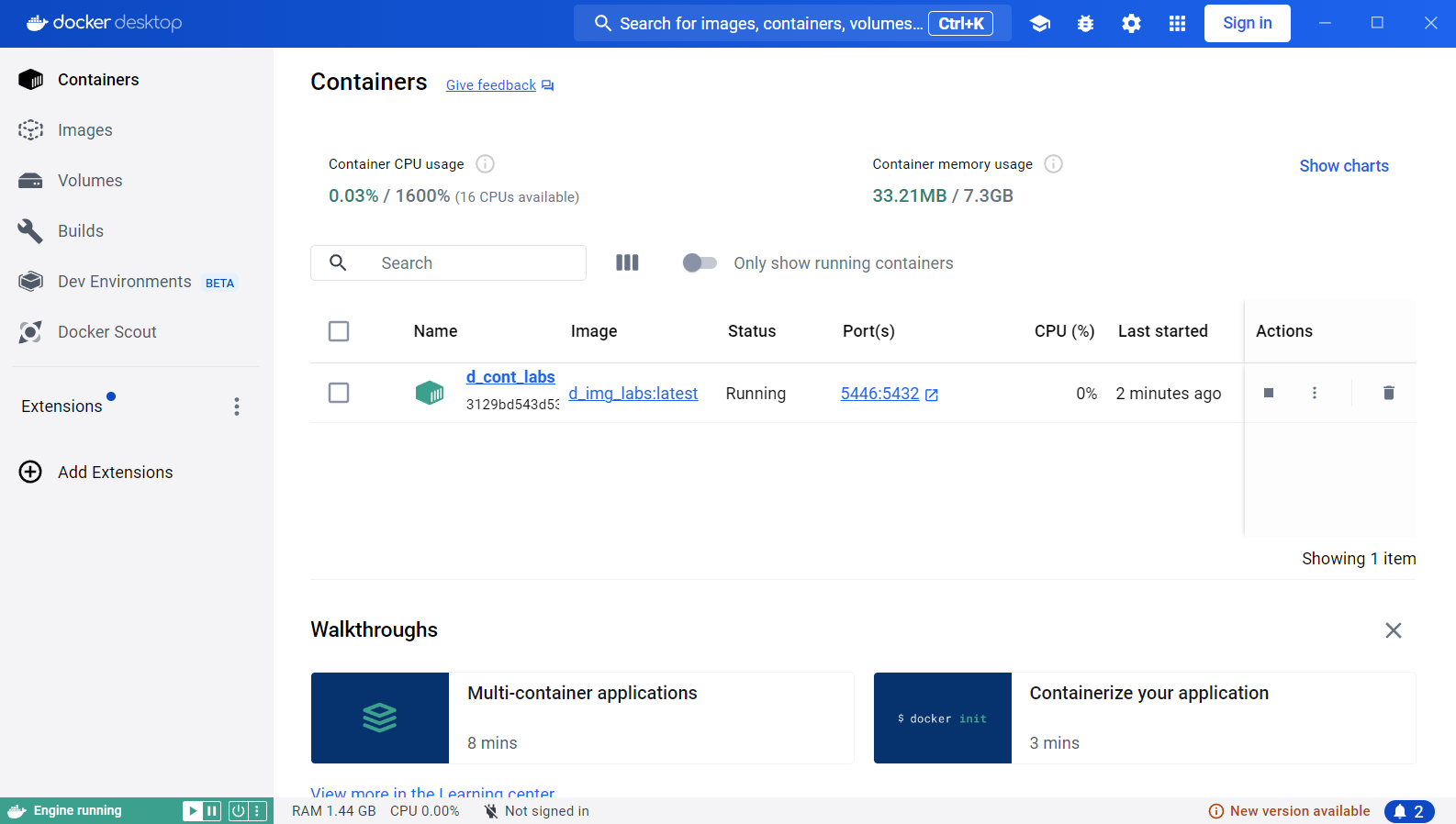
6. Разворачивание Postgres в docker-контейнере.

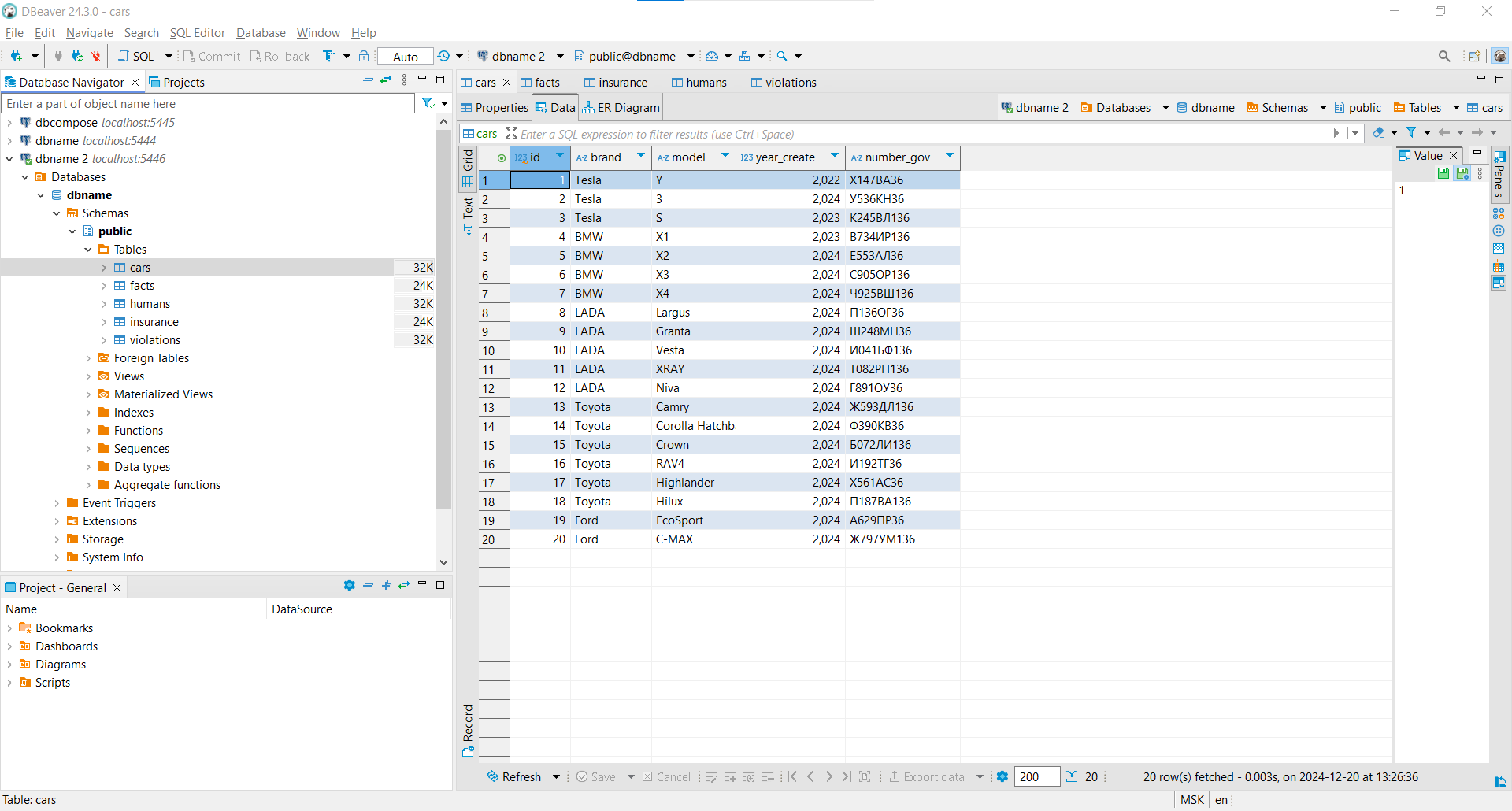
Настройки Dockerfile



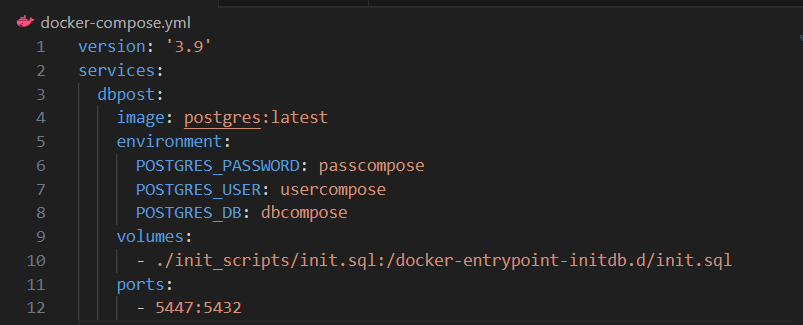




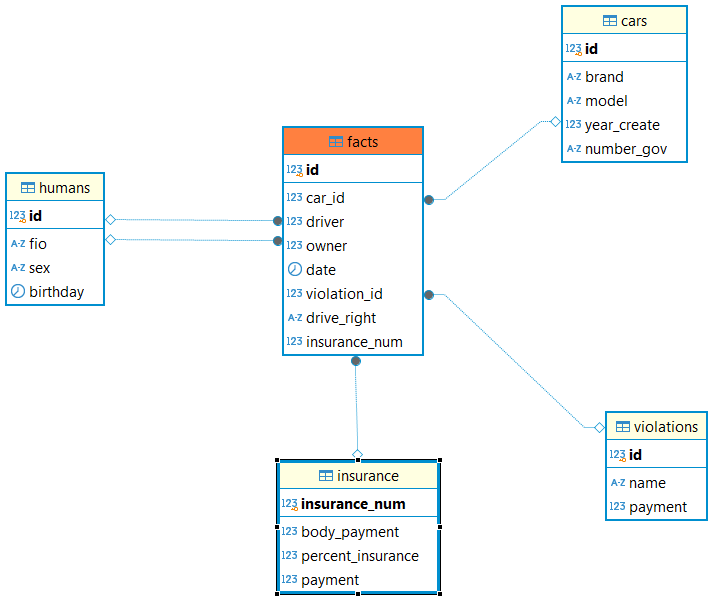




Настройки Docker compose



7. Построить графическую диаграмму, отображающую физическую структуру базы данных.



8. Описать диаграмму. Пояснить выбор сущностей, атрибутов, связей, ключей.

В качестве примера была выбрана задача построить базу данных «Учет нарушений правил дорожного движения». Для ее создания были введены следующие сущности:

* Cars (Автомобили)
* Violations (Нарушения ПДД)
* Humans (Люди)
* Insurance (Страхование)
* Facts (Факты нарушения ПДД)

Этот набор сущностей был выбран и потому, что максимально полно описывает ситуацию транспортного происшествия, а именно в ней участвуют водители, их автомобили. У каждого автомобиля должна быть страховка. При ДДП фиксируется и устанавливается факт и вид нарушения.

Cars (Автомобили):

* id (первичный ключ)
* brand (бренд)
* model (модель)
* year\_create (год выпуска)
* number\_gov (государственный номер автомобиля)

Violations (Нарушения ПДД):

* id (первичный ключ)
* name (название нарушения)
* payment (сумма штрафа)

Humans (Люди):

* id (первичный ключ)
* fio (Ф.И.О.)
* sex (пол)
* birthday (дата рождения)

Insurance (Страхование):

* insurance\_num (номер страховки, первичный ключ)
* body\_payment (стоимость страхового объекта)
* percent\_insurance (процент страховки)
* payment (сумма страховых выплат)

Facts (Факты нарушения ПДД):

* id (первичный ключ)
* car\_id (внешний ключ таблицы cars)
* driver (внешний ключ таблицы humans)
* owner (внешний ключ таблицы humans)
* date (дата ДДТ)
* violation\_id (внешний ключ таблицы violations)
* drive\_right (право на вождение, кто находился за рулем автомобиля)

insurance\_num (внешний ключ таблицы insurance)

Контрольные вопросы

1. Особенности PostgreSQL.

PostgreSQL — это мощная объектно-реляционная система баз данных с открытым исходным кодом, которая за более чем 35 лет активной разработки заслужила прочную репутацию благодаря своей надежности, функциональности и производительности. Написана на языке программирования C это обеспечивает ей высокую скорость внесения, изменения и удаления данных. Перенесен весь стандарт SQL с дополнительными функциями и дополнительным языком PL/pgSQL.

2. Что такое DDL?

Data Definition Language (DDL) группа операторов определения данных. К этим операторам относятся команды для создания, изменения и удаления таблиц, представлений и других объектов базы данных. Такие команды как CREATE TABLE, DROP TABLE, ALTER TABLE.

3. Что такое DML?

Data Manipulation Language (DML) группа операторов, манипулирования данными. К ним относятся команды для выборки строк из таблиц, вставки строк в таблицы, обновления и удаления строк. Такие команды как SELECT, INSERT INTO, UPDATE, DELETE.

4. Что такое DCL?

Data Control Language (DCL) группа операторов, управляющих привилегиями доступа к объектам базы данных. Существует несколько типов прав: SELECT, INSERT, UPDATE, DELETE, TRUNCATE, REFERENCES, TRIGGER, CREATE, CONNECT, TEMPORARY, EXECUTE и USAGE. Набор прав, применимых к определённому объекту, зависит от типа объекта (таблица, функция и т. д.). Сами команды на изменение прав объектов ALTER, GRANT, REVOKE.

5. С какой команды, как правило, начинается запрос?

Команда SELECT.

6. Какая команда используется для создания базы данных?

Команда CREATE DATABASE.

7. Какая команда используется для создания таблицы в базе данных?

Команда CREATE TABLE.

8. Ответ:

SELECT \* FROM employee WHERE mod(mod(sal, 23), 5) = 0;

SELECT \* FROM employee WHERE sal IN (1500, 3000);

9. Ответ:

SELECT \* FROM employee ORDER BY dept\_id NULLS FIRST;

10. Ответ:

'1', 'a', 'char'.

11. Ответ:

Select список полей from список таблиц having условия группировки group by поля для группировки;

Select список полей from список таблиц order by поля для сортировки union Select список полей from список таблиц order by поля для сортировки;

12. Ответ:

Select avg (advance), sum (sales) from titles group by type;