# **Установка IDE Visual Studio Code**

1. Установить приложение Docker с оф.сайта
2. Открыть IDE, в моём случае Visual SC
3. Внутри IDE загрузить 2 расширения: Docker + PostgreSQL
4. На панели слева появятся 2 иконки расширений
5. Cоздать папку, внутри которой docker файл
6. Открыть папку в IDE
7. В Explorer отобразится открытая папка внутри неё необходимо создать файл, нажав на иконку документа с +. Файл должен быть под названием dockerfile
8. Файл сохранён, иконка у файла с эмблемой докера

# **Работа с кодом для выполнения настройки контейнера и БД**

1. Ввести код в поле

# Используем официальный образ PostgreSQL

FROM postgres:17

#Версии: <https://www.postgresql.org/docs/>

# установка пакетов

RUN apt update && apt install -y \

# apt - набор утилит для установки, удаления, обновления, поиска пакетов

# APT проходит по списку репозиториев и из каждого репозитория в списке получает информацию о последних пакетах, находящихся в репозитории

# && - после обновление происходит их установка

postgresql-contrib \

# дополнительный набор расширений и утилит для PostgreSQL

vim \

&& rm -rf /var/lib/apt/lists/\*

# удаление ненужных файлов и данных, например, информация о доступных пакетах в репозиториях, временные каталоги и файлы, созданные APT во время обновления списка пакетов; очистка кэша

# rf /var/lib/apt/lists/\* - универсальный путь к каталогу

# переменные

ENV POSTGRES\_DB=Korotkova

#название своей бд

ENV POSTGRES\_USER=KorotkovaUser

#имя юзера бд

ENV POSTGRES\_PASSWORD=12345

# пароль для подключения к бд

ENV PGDATA=/var/lib/postgresql/data/pgdata

#каталог данных кластера (универсальный), где будут храниться данные PostgreSQL.

<https://postgrespro.ru/docs/postgresql/17/storage-file-layout>

RUN mkdir -p $PGDATA && chown -R postgres:postgres $PGDATA

# создание директории для хранения данных. Путь к директории хранится в PGDATA

# Копируем кастомный конфиг PostgreSQL (если нужно)

#COPY postgresql.conf /etc/postgresql/postgresql.conf

# Настраиваем параметры PostgreSQL

#RUN echo «host all all 0.0.0.0/0 md5» >> /etc/postgresql/pg\_hba.conf

# Открываем порт

EXPOSE 5432

# Устанавливаем точку монтирования для данных

VOLUME [«$PGDATA»]

# Запускаем PostgreSQL

CMD ["postgres"]

1. Сохранить
2. Открыть терминал
3. Можно поиграться с консолью

проверить на наличие активных контейнеров - *docker ps*

Если есть активные вызвать в терминале *docker stop название\_контейнера*

При необходимости можно дополнительно перезагрузить *– docker restart* *название\_контейнера*

1. Ввести команду docker build -t название- postgres .

Например,docker build -t korotkovatest-postgres .

(занимается созданием образа на основе Dockerfile)

1. Проверим, что создался, появится в списке

docker image

Ради интереса можно зайти в docker desk -> images. Там будет созданный образ

1. создать контейнер на основе образа

docker run --name название\_контейнера -e POSTGRES\_PASSWORD=пароль -d -p 5432:5432 название-образа

например, docker run --name Korotkova\_DB -e POSTGRES\_PASSWORD=12345 -d -p 5432:5432 korotkovatest-postgres

ВАЖНО: необходимо здесь явным образом задать порт, иначе подключиться к БД не получится

Доп.

Чтобы удалить:

1. Сначала остановить

docker stop название /id контейнера

1. Удалить

docker rm название/id контейнера – удаление

1. Проверить список всех активных котейнеров

docker ps

docker rmi название/id образа – удаление

проверить docker images

все команды: <https://habr.com/ru/companies/flant/articles/336654/>

# **Подключиться к БД + скрипты**

1. Открыть плагин Postgres в VSC
2. Нажать на самый первый «+» верху, появится окно add connection
3. Ввести любое название в name
4. Ввести usename, password в соответствии с данными, который были установлены в переменных

ENV POSTGRES\_USER=KorotkovaUser

ENV POSTGRES\_PASSWORD=12345

1. Нажать connect

При успехе слева появится название созданной БД

1. Навести курсор правее от названия БД, появятся значки. Нажать на тот, что открывает терминал
2. Развернуть до table и нажать на +
3. Появится поле ввода справа, ввести код

-- members - дочерняя таблица для cards

create table members(

tab\_participant int,

first\_name VARCHAR,

last\_name varchar

);

select\*from members

ALTER TABLE members

ADD PRIMARY KEY (tab\_participant); --колонка становится первичным ключём

select\*from members

INSERT INTO members VALUES (1111, 'Иванов', 'Александр');

insert into members values (2222, 'Сергеев', 'Сергей');

insert into members values (3333, 'Полякова', 'Анна');

select\*from members

# cards - дочерняя таблица для table

create table cards(

id\_card int primary key,

creator\_id int,

create\_date DATE,

--board\_id int,

--list\_id int,

title VARCHAR (100),

descriptions VARCHAR (1000),

done BOOLEAN,

participants\_id INT,

FOREIGN KEY (participants\_id) REFERENCES members (tab\_participant)

);

select\*from cards

insert into cards values (1, 2222, '2023-10-01', 'Планы', 'Описание', true, 2222);

select\*from cards

insert into cards values (2, 2222, '2023-10-11', 'Название', 'Описание', true, NULL);

select\*from cards

insert into cards values (3, 1111, '2023-10-03', 'Задача на день', 'Работать', false, 3333);

select\*from cards

insert into cards values (4, 3333, '2023-11-23', 'Написать аналитику', 'Добавить ссылку', false, 1111);

select\*from cards

# дочерняя таблица для table

create table list(

id\_list int primary key,

creator\_id int,

title VARCHAR (100)

);

select\*from list

ALTER TABLE list

ADD COLUMN id\_board INT,

ADD CONSTRAINT fk\_id\_table

FOREIGN KEY (id\_board) REFERENCES board (id\_table);

select\*from list

insert into list values (11, 1111, 'to do', 0001);

select\*from list

insert into list values (22, 1111, 'in progress', 0001);

select\*from list

insert into list values (33, 1111, 'done', 0001);

select\*from list

DELETE FROM list

WHERE id\_list = 11;

select\*from list

DELETE FROM list

WHERE id\_list = 22;

select\*from list

DELETE FROM list

WHERE id\_list = 33;

select\*from list

insert into list values (11, 1111, 'to do', 0001);

select\*from list

insert into list values (22, 1111, 'in progress', 0001);

select\*from list

insert into list values (33, 1111, 'done', 0001);

select\*from list

#главная таблица

create table board(

id\_table int primary key,

tab\_author int,

title VARCHAR (100),

--card\_id INT,

--FOREIGN KEY (card\_id) REFERENCES cards (id\_card),

--list\_id INT,

--FOREIGN KEY (list\_id) REFERENCES list (id\_list)

);

select\*from board

insert into board values (0001, 1111, 'Задачи команды');

select\*from board

insert into board values (0002, 3333, 'Доска Поляковой');

select\*from board

------------------------------------------------------------------------------------------------------------------

-- 1) Отобразить все колонки, которые были созданы Ивановым Александром id 1111

#результат - 3 колонки с id 11, 22, 33

select\*from list

where creator\_id = 1111;

# 2) Отобразить id доски с названием Доска Поляковой

#результат - доска с id 2

select id\_table

from board

where title = 'Доска Поляковой';

-- 3) Отобразить карточку, в которой creator\_id = 2222, а participants\_id = null

#результат

select\*from cards

where (creator\_id = 2222) AND (participants\_id is NULL);

-- Работа с 2-мя таблицами cards + members

-- 4) Отобразить фамилию и имя создателя карточки с id 1

-- Результат: Сергеев Сергей

SELECT cards.id\_card, members.first\_name, members.last\_name FROM cards JOIN members ON participants\_id = tab\_participant

where id\_card = 1;

-- 5) Показать названия всех колонок доски, где id не равно 0002 (=2)

select board.id\_table, list.title from list LEFT JOIN board ON id\_board = id\_table

where id\_board != 2;