**Схема БДИзображение выглядит как текст, снимок экрана, Шрифт, Печать

Автоматически созданное описание**

**Использованные типы данных в PostgreSQL:**

- smallserial, serial — целые с автоинкрементом, начинаются с 1, с ограничением сверху таким же, как у smallint и int соответственно, причём даже при ошибочном insert значение увеличивается

- smallint — целые в пределах -32768 до +32767

- int — целые в пределах -2147483648 до +2147483647

- bigint - целые в пределах -9223372036854775808 до +9223372036854775807

- numeric(n, m) — числа с m цифр после запятой, всего цифр в числе n, т.е. до запятой n - m цифр

- boolean - логический тип, токо true или false

- varchar(n) — строка с максимальной длиной n

- text — строка произвольной длины

- timestamptz — дата и время с часовым поясом, секунды с 6 цифрами после запятой

**1. Enterprises (Предприятия — клиенты)**

- **Поля**:

- id\_enterprise (PK, smallserial)

- title (varchar(100)) — название предприятия

- type\_property (varchar(50)) — тип собственности предприятия

- address (varchar(150))

- tel (varchar(11))

**2. Employees (Сотрудники)**

- **Поля**:

- id\_employee (PK, smallserial)

- id\_post (FK, smallint) — должность сотрудника

**3. Posts (Должности)**

- **Поля**:

- id\_post (PK, smallserial)

- post\_name (varchar(50))

- salary (numeric(8, 2)) — зарплата, связанная с данной должностью

**4. Users (Пользователи)**

Либо id\_employee, либо id\_enterprise равен null, т.е. юзер — ето либо сотрудник, либо клиент

- **Поля**:

- id\_user (PK, smallserial)

- id\_employee (FK, smallint)

- id\_enterprise (FK, smallint)

**5. Messages (Сообщения)**

- **Поля**:

- id\_message (PK, serial)

- id\_sender (FK, smallint) — внешний ключ на Users (id\_user)

- id\_receiver (FK, smallint) — внешний ключ на Users (id\_user)

- message\_date (timestamptz)

- message\_content (text)

**6. Portfolios (Портфели клиентов)**

Хоть и по тз у одного клиента может быть токо один портфель, спроектировано здесь на множество портфелей

- **Поля**:

- id\_portfolio (PK, smallserial)

- id\_employee (FK, smallint) — id сотрудника, ответственного за портфель в данный момент времени, в начале равен null, пока какой-либо сотрудник не возьмётся за данный портфель

- id\_enterprise (FK, smallint) — id клиента, которому принадлежит данный портфель

- balance (numeric(20, 5)) — сумма вложенных средств + прибыль/убыток с ценных бумаг, автоматически пересчитывается при каждом изменении котировок ценных бумаг и при каждом пополнении/снятии средств клиентом

- deposition (numeric(20, 5)) — сумма вложенных средств, автоматически пересчитывается при каждом пополнении/снятии средств клиентом

- account\_number (numeric(21, 0)) — номер лицевого счёта, должно генерироваться автоматически и быть уникальным для каждого портфеля (в бд не реализовано)

**7. Balances History (История изменения баланса каждого портфеля)**

Каждый месяц нужно добавлять записи — баланс в данный момент времени, в бд не реализовано

- **Поля**:

- id\_portfolio (PK, FK, smallint)

- balance\_date (PK, timestamptz) — дата и время зафиксированного баланса данного портфеля

- balance (numeric(20, 5))

**8. Operations History (История операций каждого портфеля клиентом)**

- **Поля**:

- id\_oper (PK, serial)

- id\_portfolio (FK, smallint)

- oper\_type (boolean) — true = пополнение, false = снятие

- oper\_status (boolean): default = true — true = успешная операция, false = отменённая операция (с ошибкой)

- oper\_date (timestamptz): default = текущая дата и время

- amount (numeric(20, 5)) — сумма операции

**9. Securities (Ценные бумаги)**

Причём здесь так же есть рубли (id\_securitie = 36), необходимые для хранения свободных средств портфеля

- **Поля**:

- id\_securitie (PK, smallserial)

- id\_catalog (FK, smallint)

- id\_add\_info (FK, smallint) — null для всех, кроме облигаций (id\_catalog = 2)

- sec\_name (varchar(100))

- ticker (varchar(100)) — тикер ценной бумаги

- quotation (numeric(20, 5)) — котировка (цена) ценной бумаги в данный момент времени, для id\_securitie = 36 всегда равна 1

- description (text) — описание данной ценной бумаги, для некоторых null

- icon (varchar(50)) — название (с расширением) иконки для ценной бумаги, файл хранится на гите в app/front\_app/static/logo\_org

**10. Catalogs (Каталоги — виды ценных бумаг)**

- **Поля**:

- id\_catalog (PK, smallserial)

- catalog\_name (varchar(100))

**11. Additional Info (Дополнительная информация — токо для облигаций)**

- **Поля**:

- id\_add\_info (PK, smallserial)

- yield\_close (varchar(20)) — доходность по погашению

- closing\_date (timestamptz) — дата погашения

- coupon\_payment\_date (timestamptz) — дата выплаты купона

- coupon (varchar(50)) — вид купона

- coupon\_value (numeric(20, 5)) — величина купона

- nominal (numeric(20, 5)) — номинал

- number\_payments\_per\_year (bigint) — количество выплат в год

- depreciation (varchar(10)) — амортизация

**12. Quotations History (История котировок)**

Автоматически добавляются записи при каждом изменении котировок

- **Поля**:

- id\_securitie (PK, FK, smallint)

- sec\_date (PK, timestamptz) — дата зафиксированной котировки для данной ценной бумаги

- quotation (numeric(20, 5))

**13. Requests (Запросы)**

Информация о запросах сотрудника на покупку или продажу ценных бумаг

- **Поля**:

- id\_request (PK, serial)

- id\_portfolio (FK, smallint)

- id\_employee (FK, smallint): default = id сотрудника, сделавшего данный запрос

- id\_securitie (FK, smallint)

- req\_type (boolean) — true = покупка, false = продажа

- req\_status (boolean): default = true — true = успешный запрос, false = отменённый запрос (с ошибкой)

- quantity (bigint) — количество ценной бумаги

- req\_date (timestamptz): default = текущая дата и время

- commission (numeric(20, 5)): default = 0 — равна 0 для покупок, считается автоматически для продаж

- total\_price (numeric(20, 5)): default = 0 — общая цена запроса, считается автоматически

**14. Portfolio to Securitie**

Количество и средняя цена каждой ценной бумаги у каждого портфеля в данный момент времени, а также свободные средства у каждого портфеля

- **Поля**:

- id\_portfolio (PK, FK, smallint)

- id\_securitie (PK, FK, smallint)

- total\_quantity (numeric(20, 5)) — количество данной ценной бумаги, а для id\_securitie = 36 — свободные средства у данного портфеля

- avg\_quotation (numeric(20, 5)) — средняя цена данной ценной бумаги для данного портфеля, перечитывается автоматически при каждой покупке/продаже ценной бумаги данного портфеля, для id\_securitie = 36 всегда равна 1

**15. Queue (Очередь)**

Очередь купленных (и не проданных на данный момент) ценных бумаг у каждого портфеля с количеством и котировкой в момент времени покупки, автоматически добавляются/удаляются записи для корректного подсчёта средней цены ценных бумаг

- **Поля**:

- id\_portfolio (PK, FK, smallint)

- id\_securitie (PK, FK, smallint)

- queue\_date (PK, timestamptz)

- quantity (bigint)

- quotation (numeric(20, 5))

Так же в таблице auth\_user, созданной Django, добавили столбцы id\_user и patronymic

id\_user — для связки 1:1 с таблицей Users

patronymic — для хранения отчества

**Триггеры**

Триггер insert\_quotations\_trigger, установленный на любое изменение строки таблицы securities, записывает в таблицу quotations\_history новые котировки каждой ценной бумаги с текущей датой и временем

Триггер change\_avg\_trigger, установленный на удаление, вставку или изменение любой строки в таблицу queue, пересчитывает среднюю цену данной ценной бумаги avg\_quotation в таблице portfolio\_to\_securitie у данного портфеля

Триггер change\_deposition\_balance\_funds\_trigger, установленный на вставку любой строки в таблицу operations\_history, если oper\_type = true, добавляет amount к deposition и balance в таблице portfolios данного портфеля и добавляет amount к свободным средствам данного портфеля, если oper\_type = false, то проверяет, хватает ли свободных средств для вывода amount рублей у данного портфеля, если хватает, то отнимает amount от deposition и balance в таблице portfolios данного портфеля и отнимает amount от свободных средств для данного портфеля, если не хватает свободных средств, то устанавливает oper\_status на false

Триггер change\_portfolio\_to\_securitie\_queue\_trigger, установленный на вставку любой строки в таблице requests, считает сумму заявки total\_price и устанавливает id\_employee, отвечающий за данный портфель, если req\_type = true, то проверяет, хватает ли свободных средств у данного портфеля для покупки на total\_price, если хватает, то прибавляет количество quantity данной ценной бумаги к total\_quantity в таблице portfolio\_to\_securitie для данного портфеля, отнимает total\_price от свободных средств данного портфеля и вставляет в таблицу queue строку с req\_date, quantity и котировкой данной ценной бумаги на данный момент времени для данного портфеля, если свободных средств не хватает, то устанавливает req\_status на false, если req\_type = false, то проверяет, есть ли в данный момент времени у данного портфеля данных ценных бумаг в количестве quantity, если есть, то подсчитывает комиссию, попутно удаляя из таблицы queue строки с самыми старыми записями с данной ценной бумагой для данного портфеля до тех пор, пока не наберётся в сумме данных ценных бумаг в количестве quantity, отнимает quantity от total\_quantity данной ценной бумаги в таблице portfolio\_to\_securitie для данного портфеля и добавляет total\_priceк свободным средствам данного портфеля, вычитая посчитанную комиссию, если нету у данного портфеля данных ценных бумаг в количестве quantity, то устанавливает req\_status на false