Федеральное государственное бюджетное образовательное учреждение высшего

образования

«Чувашский государственный университет им. И.Н. Ульянова»

Кафедра вычислительной техники

Лабораторная работа № 3

Реализация РБД

Вариант: Инвестор

Выполнил: Назаров Ярослав Сергеевич

студент группы ИВТ-42-23

Проверил: кандидат технических наук

Ржавин Вячеслав Валентинович

Чебоксары, 2025

**Состав базы данных**

|  |  |  |
| --- | --- | --- |
| **№** | **Название таблицы** | **Названия полей** |
| 1 | ИНВЕСТОР | #И, ФИО, ДатаРождения, КонтактныйТелефон, ЭлектроннаяПочта |
| 2 | БРОКЕР | #БР, Наименование, Лицензия, КонтактныеДанные |
| 3 | СЧЁТ | #С, НомерСчета, ТипСчета, ДатаОткрытия, Статус, #И(FK), #БР(FK) |
| 4 | АКТИВ | #А, ЦенаЗаЕдиницу, Тикер, Наименование, ТипАктива, ВалютаНоминала |
| 5 | ТРАНЗАКЦИЯ | #Т, ДатаВремя, ТипОперации, Количество, СуммаТранзакции, Валюта, #С(FK), #А(FK) |
| 6 | СВЯЗЬ БРОКЕР-ИНВЕСТОР | #СБИ, #БР(FK), #И(FK) |
| 7 | СВЯЗЬ СЧЁТ-АКТИВ | #ССА, #С(FK), #А(FK), Количество |

**Описание характеристик атрибутов**

**Инвестор**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **№** | **Имя атрибута** | **А1** | **А2** | **А3** | **А4** | **А5** | **А6** | **А7** |
| 1 | #И | Счётчик | — | — | Номер инвестора | — | Да | Да |
| 2 | ФИО | Текстовый | 80 | — | Полное имя | — | Да | Нет |
| 3 | ДатаРождения | Дата/Время | — | Краткая дата | Дата рождения | — | Да | Нет |
| 4 | КонтактныйТелефон | Текстовый | 20 | — | Телефон | — | Да | Да |
| 5 | ЭлектроннаяПочта | Текстовый | 50 | — | Email | — | Нет | Да |

**Брокер**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **№** | **Имя атрибута** | **А1** | **А2** | **А3** | **А4** | **А5** | **А6** | **А7** |
| 1 | #БР | Счётчик | — | — | Номер брокера | — | Да | Да |
| 2 | Наименование | Текстовый | 100 | — | Название компании | — | Да | Нет |
| 3 | Лицензия | Текстовый | 30 | — | Номер лицензии | — | Да | Да |
| 4 | КонтактныеДанные | Текстовый | 200 | — | Контакты | — | Нет | Нет |

**Счёт**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **№** | **Имя атрибута** | **А1** | **А2** | **А3** | **А4** | **А5** | **А6** | **А7** |
| 1 | #С | Счётчик | — | — | Номер счёта | — | Да | Да |
| 2 | НомерСчета | Текстовый | 20 | — | Уникальный номер | — | Да | Да |
| 3 | ТипСчета | Текстовый | 20 | — | Тип счёта | — | Да | Нет |
| 4 | ДатаОткрытия | Дата/Время | — | Краткая дата | Дата открытия | CurrentDate | Да | Нет |
| 5 | Статус | Текстовый | 15 | — | Статус | 'Активен' | Да | Нет |
| 6 | #И(FK) | Числовой | — | — | Инвестор (FK) | — | Да | Да |
| 7 | #БР(FK) | Числовой | — | — | Брокер (FK) | — | Да | Да |

**Актив**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **№** | **Имя атрибута** | **А1** | **А2** | **А3** | **А4** | **А5** | **А6** | **А7** |
| 1 | #А | Счётчик | — | — | Номер актива | — | Да | Да |
| 2 | ЦенаЗаЕдиницу | Числовой | — | Денежный | Цена за единицу | — | Да | Нет |
| 3 | Тикер | Текстовый | 10 | — | Торговый код | — | Да | Да |
| 4 | Наименование | Текстовый | 100 | — | Название актива | — | Да | Нет |
| 5 | ТипАктива | Текстовый | 20 | — | Тип актива | — | Да | Нет |
| 6 | ВалютаНоминала | Текстовый | 5 | — | Валюта | — | Да | Нет |

**Транзакция**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **№** | **Имя атрибута** | **А1** | **А2** | **А3** | **А4** | **А5** | **А6** | **А7** |
| 1 | #Т | Счётчик | — | — | Номер транзакции | — | Да | Да |
| 2 | ДатаВремя | Дата/Время | — | Полная дата | Дата и время | CurrentDate | Да | Да |
| 3 | ТипОперации | Текстовый | 20 | — | Тип операции | — | Да | Нет |
| 4 | Количество | Числовой | — | — | Количество | 1 | Да | Нет |
| 5 | СуммаТранзакции | Числовой | — | Денежный | Сумма | — | Да | Нет |
| 6 | Валюта | Текстовый | 5 | — | Валюта | — | Да | Нет |
| 7 | #С(FK) | Числовой | — | — | Счёт (FK) | — | Да | Да |
| 8 | #А(FK) | Числовой | — | — | Актив (FK) | — | Нет | Да |

**Связь Брокер-Инвестор**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **№** | **Имя атрибута** | **А1** | **А2** | **А3** | **А4** | **А5** | **А6** | **А7** |
| 1 | #СБИ | Счётчик | — | — | Номер связи | — | Да | Да |
| 2 | #БР(FK) | Числовой | — | — | Брокер (FK) | — | Да | Да |
| 3 | #И(FK) | Числовой | — | — | Инвестор (FK) | — | Да | Да |

**Связь Актив-Счёт**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **№** | **Имя атрибута** | **А1** | **А2** | **А3** | **А4** | **А5** | **А6** | **А7** |
| 1 | #ССА | Счётчик | — | — | Номер связи | — | Да | Да |
| 2 | #С(FK) | Числовой | — | — | Счёт (FK) | — | Да | Да |
| 3 | #А(FK) | Числовой | — | — | Актив (FK) | — | Да | Да |
| 4 | Количество | Числовой | — | — | Количество актива | — | Да | Нет |

Где:

* Тип данных (А1).
* Размер поля (А2).
* Формат вывода значений данного поля (А3).
* Подпись поля в форме (А4).
* Значение по умолчанию (А5).
* Обязательное поле (А6).
* Индексированное поле (А7).

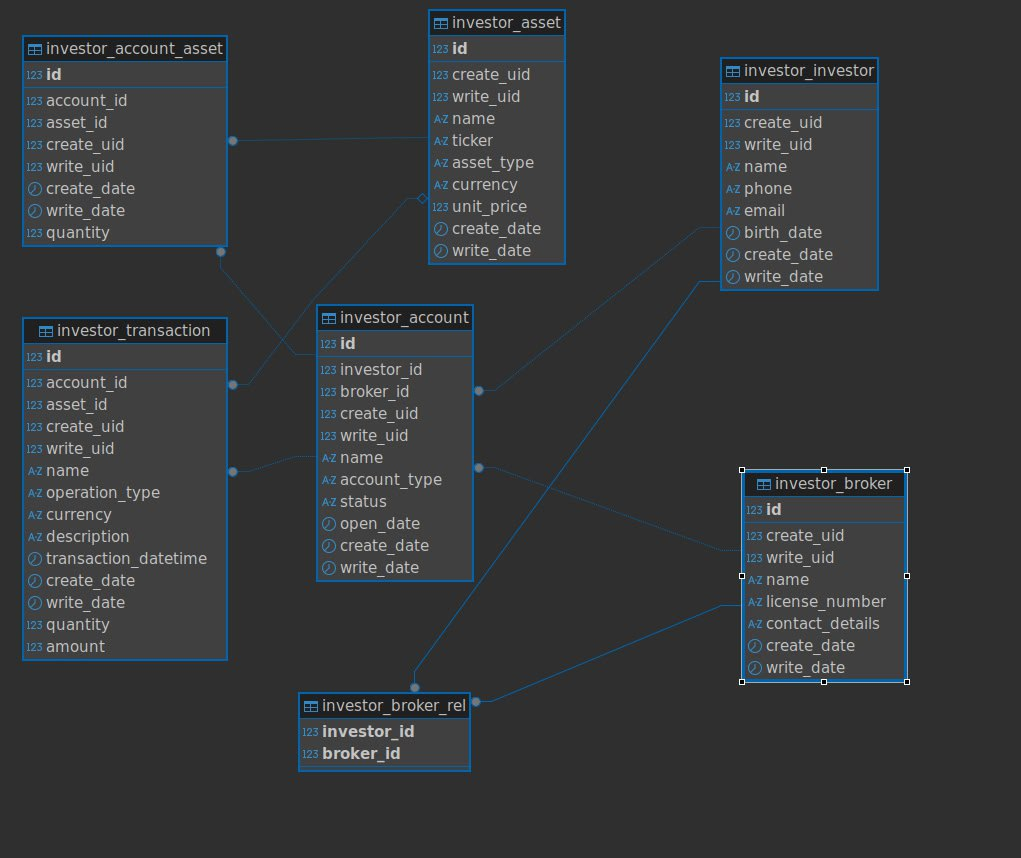
**Правила валидации**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№** | **Отношение** | **Атрибут** | **Условие на значение** | **Сообщение об ошибке** |
| 1 | ИНВЕСТОР | ДатаРождения | < Date() | Дата рождения не может быть в будущем |
| 2 | ИНВЕСТОР | ЭлектроннаяПочта | LIKE "\*@\*.\*" | Некорректный формат email |
| 3 | СЧЁТ | НомерСчета | UNIQUE | Номер счёта должен быть уникальным |
| 4 | СЧЁТ | ТипСчета | "ИИС" OR "Брокерский" OR "Депозитарный" | Недопустимый тип счёта |
| 5 | АКТИВ | Тикер | LEN([Тикер]) >= 3 | Тикер должен содержать ≥3 символов |
| 6 | ТРАНЗАКЦИЯ | Количество | > 0 | Количество должно быть положительным |
| 7 | ТРАНЗАКЦИЯ | СуммаТранзакции | > 0 | Сумма должна быть положительной |
| 8 | ИНВЕСТОР | КонтактныйТелефон | Соответствие формату +7 (XXX) XXX-XX-XX | Неверный формат телефона. Пример: +7 (999) 123-45-67 |
| 9 | АКТИВ | Тикер | Все символы в верхнем регистре | Тикер должен содержать 3 или более символов в верхнем регистре. |
| 10 | СВЯЗЬ СЧЁТ-АКТИВ | #С(FK), #А(FK) | UNIQUE (по паре полей) | Этот актив уже существует на данном счете. |
| 11 | СЧЁТ | ДатаОткрытия | ≤ Date() | Дата открытия не может быть в будущем. |
| 12 | ТРАНЗАКЦИЯ | ДатаВремя | ≤ Now() | Дата и время транзакции не могут быть в будущем. |
| 13 | БРОКЕР | Лицензия | UNIQUE | Номер лицензии должен быть уникальным. |
| 14 | ТРАНЗАКЦИЯ | Актив (#А) | NOT NULL для операций "Покупка" или "Продажа" | Для операций покупки или продажи необходимо указать актив. |
| 15 | СВЯЗЬ СЧЁТ-АКТИВ | Количество | >= 0 | Количество не может быть отрицательным |
| 16 | ИНВЕСТОР | ЭлектроннаяПочта | UNIQUE | Электронная почта должна быть уникальной. |
| 17 | ИНВЕСТОР | КонтактныйТелефон | UNIQUE | Контактный телефон должен быть уникальным. |

**Описание подстановок**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Таблица** | **Исходное поле (FK)** | **Поле подстановки** | **Тип элемента управления** | **Тип источника строк** | **Источник строк (SQL-запрос)** |  |
| СЧЁТ | #И(FK) | ФИО | Поле со списком | Таблица или запрос | SELECT [#И], [ФИО] FROM ИНВЕСТОР ORDER BY [#И]; |  |
| СЧЁТ | #БР(FK) | Наименование | Поле со списком | Таблица или запрос | SELECT [#БР], [Наименование] FROM БРОКЕР ORDER BY [#БР]; |  |
| ТРАНЗАКЦИЯ | #С(FK) | НомерСчета | Поле со списком | Таблица или запрос | SELECT [#С], [НомерСчета] FROM СЧЁТ ORDER BY [#С]; |  |
| ТРАНЗАКЦИЯ | #А(FK) | Тикер | Поле со списком | Таблица или запрос | SELECT [#А], [Тикер], [Наименование] FROM АКТИВ ORDER BY [#А]; |  |
| СВЯЗЬ БРОКЕР-ИНВЕСТОР | #БР(FK) | Наименование | Поле со списком | Таблица или запрос | SELECT [#БР], [Наименование] FROM БРОКЕР ORDER BY [#БР]; |  |
| СВЯЗЬ БРОКЕР-ИНВЕСТОР | #И(FK) | ФИО | Поле со списком | Таблица или запрос | SELECT [#И], [ФИО] FROM ИНВЕСТОР ORDER BY [#И]; |  |
| СВЯЗЬ СЧЁТ-АКТИВ | #С(FK) | НомерСчета | Поле со списком | Таблица или запрос | SELECT [#С], [НомерСчета] FROM СЧЁТ ORDER BY [#С]; |  |
| СВЯЗЬ СЧЁТ-АКТИВ | #А(FK) | Тикер | Поле со списком | Таблица или запрос | SELECT [#А], [Тикер], [Наименование] FROM АКТИВ ORDER BY [#А]; |  |

**Получившаяся база данных в DBeaver:**



**Вывод:**

В ходе выполнения данной лабораторной работы была успешно решена задача по реализации реляционной базы данных на тему "Инвестор". На основе спроектированной структуры, включающей семь взаимосвязанных таблиц, были созданы модели данных с использованием платформы Odoo. Были детально определены атрибуты для каждой таблицы, включая их типы данных, размеры, форматы и значения по умолчанию.

Классы в Odoo:

from odoo import models, fields, api

from odoo.exceptions import ValidationError

import re

class Investor(models.Model):

    \_name = 'investor.investor'

    \_description = 'Инвестор'

    \_table = 'investor\_investor'

    name = fields.Char(string="ФИО", required=True, index=True)

    birth\_date = fields.Date(string="Дата Рождения", required=True)

    phone = fields.Char(string="Контактный Телефон", required=True, index=True)

    email = fields.Char(string="Электронная Почта", required=True, index=True)

    account\_ids = fields.One2many('investor.account', 'investor\_id', string="Счета")

    broker\_ids = fields.Many2many(

        'investor.broker',

        'investor\_broker\_rel',

        'investor\_id',

        'broker\_id',

        string="Брокеры"

    )

    \_sql\_constraints = [

        ('phone\_uniq', 'unique (phone)', 'Контактный телефон должен быть уникальным.'),

        ('email\_uniq', 'unique (email)', 'Электронная почта должна быть уникальной.')

    ]

    @api.constrains('email')

    def \_check\_email\_format(self):

        for record in self:

            if record.email and not re.match(r"[^@]+@[^@]+\.[^@]+", record.email):

                raise ValidationError("Некорректный формат электронной почты.")

    @api.constrains('birth\_date')

    def \_check\_birth\_date(self):

        for record in self:

            if record.birth\_date and record.birth\_date > fields.Date.today():

                 raise ValidationError("Дата рождения не может быть в будущем.")

    @api.constrains('phone')

    def \_check\_phone\_format(self):

        for record in self:

            if record.phone and not re.match(r"\+7\s\(\d{3}\)\s\d{3}-\d{2}-\d{2}", record.phone):

                raise ValidationError("Неверный формат телефона. Пример: +7 (999) 123-45-67")

class Account(models.Model):

    \_name = 'investor.account'

    \_description = 'Счет'

    name = fields.Char(string="Номер Счета", required=True, index=True)

    account\_type = fields.Selection([

        ('iis', 'ИИС'),

        ('broker', 'Брокерский'),

        ('depository', 'Депозитарный')

    ], string="Тип Счета", required=True)

    open\_date = fields.Date(string="Дата Открытия", required=True, default=fields.Date.context\_today)

    status = fields.Selection([

        ('active', 'Активен'),

        ('closed', 'Закрыт'),

        ('blocked', 'Заблокирован')

    ], string="Статус", required=True, default='active')

    investor\_id = fields.Many2one('investor.investor', string="Инвестор", required=True, index=True)

    broker\_id = fields.Many2one('investor.broker', string="Брокер", required=True, index=True)

    transaction\_ids = fields.One2many('investor.transaction', 'account\_id', string="Транзакции")

    asset\_line\_ids = fields.One2many('investor.account.asset', 'account\_id', string="Активы на счете")

    \_sql\_constraints = [

        ('name\_uniq', 'unique (name)', 'Номер счета должен быть уникальным.')

    ]

    @api.constrains('open\_date')

    def \_check\_open\_date(self):

        for record in self:

            if record.open\_date and record.open\_date > fields.Date.today():

                raise ValidationError("Дата открытия не может быть в будущем.")

class Asset(models.Model):

    \_name = 'investor.asset'

    \_description = 'Актив'

    name = fields.Char(string="Наименование", required=True)

    ticker = fields.Char(string="Тикер", required=True, index=True)

    asset\_type = fields.Selection([

        ('stock', 'Акция'),

        ('bond', 'Облигация'),

        ('currency', 'Валюта'),

        ('fund', 'Фонд')

    ], string="Тип Актива", required=True)

    currency = fields.Selection([

        ('RUB', 'RUB'),

        ('USD', 'USD'),

        ('EUR', 'EUR'),

        ('CNY', 'CNY')

    ], string="Валюта Номинала", required=True, default='RUB')

    unit\_price = fields.Float(string="Цена за Единицу", digits='Product Price', required=True)

    @api.constrains('ticker')

    def \_check\_ticker(self):

        for record in self:

            if len(record.ticker) < 3 or not record.ticker.isupper():

                raise ValidationError("Тикер должен содержать 3 или более символов в верхнем регистре.")

class AccountAsset(models.Model):

    \_name = 'investor.account.asset'

    \_description = 'Актив на Счете'

    \_rec\_name = 'asset\_id'

    account\_id = fields.Many2one('investor.account', string="Счет", required=True, ondelete='cascade', index=True)

    asset\_id = fields.Many2one('investor.asset', string="Актив", required=True, ondelete='cascade', index=True)

    quantity = fields.Float(string="Количество", required=True, default=0.0)

    \_sql\_constraints = [

        ('account\_asset\_uniq', 'unique (account\_id, asset\_id)', 'Этот актив уже существует на данном счете.')

    ]

    @api.constrains('quantity')

    def \_check\_quantity(self):

        for record in self:

            if record.quantity < 0:

                raise ValidationError("Количество актива не может быть отрицательным.")

class Transaction(models.Model):

    \_name = 'investor.transaction'

    \_description = 'Транзакция'

    \_order = 'transaction\_datetime desc'

    name = fields.Char(string="Описание", compute='\_compute\_display\_name', required=True, store=True)

    transaction\_datetime = fields.Datetime(string="Дата и Время", required=True, default=fields.Datetime.now, index=True)

    operation\_type = fields.Selection([

        ('buy', 'Покупка'),

        ('sell', 'Продажа'),

        ('deposit', 'Зачисление'),

        ('withdrawal', 'Списание'),

        ('commission', 'Комиссия')

    ], string="Тип Операции", required=True)

    quantity = fields.Float(string="Количество", default=1.0, required=True)

    amount = fields.Float(string="Сумма Транзакции", required=True)

    currency = fields.Selection([

        ('RUB', 'RUB'), ('USD', 'USD'), ('EUR', 'EUR')

    ], string="Валюта", required=True, default='RUB')

    account\_id = fields.Many2one('investor.account', string="Счет", required=True, index=True)

    asset\_id = fields.Many2one('investor.asset', string="Актив", index=True)

    description = fields.Text(string="Дополнительные детали")

    @api.depends('operation\_type', 'asset\_id.name', 'account\_id.name')

    def \_compute\_display\_name(self):

        for trans in self:

            name\_parts = [dict(self.\_fields['operation\_type'].selection).get(trans.operation\_type)]

            if trans.asset\_id:

                name\_parts.append(trans.asset\_id.name)

            if trans.account\_id:

                name\_parts.append(f"({trans.account\_id.name})")

            trans.name = " ".join(filter(None, name\_parts))

    @api.constrains('quantity', 'amount')

    def \_check\_positive\_values(self):

        for record in self:

            if record.quantity <= 0:

                raise ValidationError("Количество должно быть положительным.")

            if record.amount <= 0:

                raise ValidationError("Сумма транзакции должна быть положительной.")

    @api.constrains('transaction\_datetime')

    def \_check\_transaction\_datetime(self):

        for record in self:

            if record.transaction\_datetime and record.transaction\_datetime > fields.Datetime.now():

                raise ValidationError("Дата и время транзакции не могут быть в будущем.")

    @api.constrains('operation\_type', 'asset\_id')

    def \_check\_asset\_for\_buy\_sell(self):

        for record in self:

            if record.operation\_type in ['buy', 'sell'] and not record.asset\_id:

                raise ValidationError("Для операций покупки или продажи необходимо указать актив.")

class Broker(models.Model):

    \_name = 'investor.broker'

    \_description = 'Брокер'

    name = fields.Char(string="Наименование", required=True)

    license\_number = fields.Char(string="Лицензия", required=True, index=True)

    contact\_details = fields.Text(string="Контактные Данные")

    \_sql\_constraints = [

        ('license\_number\_uniq', 'unique (license\_number)', 'Номер лицензии должен быть уникальным.')

    ]