Федеральное государственное бюджетное образовательное учреждение высшего образования

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

Кафедра вычислительной техники

Лабораторная работа №5 “Формы” по дисциплине Базы данных

Вариант: Инвестор

Выполнил:

студент гр.ИВТ-42-23,

Назаров Я. С.

Проверил: старший

преподаватель Марков А.В.

Чебоксары, 2025

**Реестр представлений модуля «Инвестиционный модуль»**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№** | **Идентификатор (XML ID)** | **Тип представления** | **Модель (res\_model)** | **Назначение и Примечания** |
| 1 | view\_investor\_form | Форма (Form) | investor.investor | Форма с вкладками для ведения инвестора; вкладки: «Счета» (One2many account\_ids, только чтение), «Брокеры» (many2many\_tags). |
| 2 | view\_investor\_tree | Список (Tree) | investor.investor | Реестр инвесторов с основными контактами. |
| 3 | view\_account\_form | Форма (Form) | investor.account | Иерархическая форма с вкладками: One2many asset\_line\_ids с редактированием внизу (editable="bottom"); список transaction\_ids только чтение. |
| 4 | view\_account\_tree | Список (Tree) | investor.account | Реестр счетов с типом и статусом. |
| 5 | view\_asset\_form | Форма (Form) | investor.asset | Форма справочника активов с ценой и валютой. |
| 6 | view\_asset\_tree | Список (Tree) | investor.asset | Реестр активов. |
| 7 | view\_broker\_form | Форма (Form) | investor.broker | Форма справочника брокеров. |
| 8 | view\_broker\_tree | Список (Tree) | investor.broker | Реестр брокеров. |
| 9 | view\_transaction\_form | Форма (Form) | investor.transaction | Форма ввода транзакции; поле asset\_id скрывается при несоответствии типу операции; содержит вычисляемое поле name. |
| 10 | view\_transaction\_tree | Список (Tree) | investor.transaction | Реестр транзакций; применено условное форматирование строк (decoration-...). |
| 11 | view\_transaction\_pivot | Сводная (Pivot) | investor.transaction | Итоговый анализ транзакций по типам операций; мера — amount. |
| 12 | view\_transaction\_graph | График (Graph) | investor.transaction | Аналитика транзакций (pie) по типам операций, мера — amount. |
| 13 | view\_users\_form\_inherit | Наследуемая форма (Form inherit) | res.users | Вкладка «Инвестиции» в форме пользователей; радиовыбор роли, динамическая видимость групп полей. |
| 14 | view\_account\_request\_form | Форма (Form) | investor.account.request | Мастер-заявка на открытие счета; кнопки «Одобрить/Отклонить», ограничения видимости, только чтение ключевых полей. |
| 15 | view\_income\_report\_wizard\_form | Форма (Form, Wizard) | investor.income\_report.wizard | Диалоговое окно мастера параметров отчета, кнопка вызова метода action\_generate\_report. |

Примечания к структурам:

- **Иерархические формы:** `view\_account\_form` — главная форма счета с подчиненной строкой активов (`investor.account.asset`) через `One2many` и встроенное редактирование.

- **Формы с вкладками:** `view\_investor\_form`, `view\_account\_form` используют `<notebook>` и `<page>`.

- **Аналитические представления:** `view\_transaction\_pivot`, `view\_transaction\_graph`.

- **Вычисляемые поля:** в форме транзакции отображается поле `name` (`compute='\_compute\_name'` в модели `investor.transaction`).

**Примеры реализации интерфейса**

[Скриншот 1: Главная форма «Инвестиционный счет» с вкладками и списком активов]

[Скриншот 2: Аналитическое представление (Pivot/Graph), показывающее распределение сумм транзакций по типам операций]

**Описание свойств представлений**

**Анализ XML-структуры (Архивариус)**

Ниже приведен полный XML-код формы счета из `view\_account\_form` (фрагмент от `<form>` до `</form>`):

```…investor/investor/views/account\_views.xml

<form string="Счет">

<sheet>

<group>

<group>

<field name="name"/>

<field name="investor\_id"/>

<field name="broker\_id"/>

</group>

<group>

<field name="account\_type"/>

<field name="open\_date"/>

<field name="status"/>

</group>

</group>

<notebook>

<page string="Активы на счете">

<field name="asset\_line\_ids">

<tree editable="bottom">

<field name="asset\_id"/>

<field name="quantity"/>

</tree>

</field>

</page>

<page string="История Транзакций">

<field name="transaction\_ids" readonly="1"/>

</page>

</notebook>

</sheet>

</form>

```

**Ключевые особенности:**

- Вкладки с разбивкой на подчиненные сущности.

- Встроенное редактирование строк One2many через `<tree editable="bottom">`.

- Исторические записи транзакций доступны только для чтения.

**Описание ключевых атрибутов (Свойств)**

Ниже приведены атрибуты, фактически используемые в XML-представлениях и моделях модуля:

- `**string**` (Заголовок): текстовый заголовок для форм, страниц, полей и представлений.

- `**readonly**` (Блокировка): ограничивает редактирование поля; встречается как статически (`readonly="1"`), так и с выражением (`readonly="id!=False"`).

- `**required**` (Обязательность): указано на уровне модели (Python) для полей, определяет необходимость заполнения.

- `**invisible**` (Видимость): скрывает поля по условию в XML, напр. `invisible="status!='draft'"`, `invisible="operation\_type not in ['buy', 'sell']"`, а также на группах в унаследованной форме `res.users`.

- `**widget**` (Виджет): `phone`, `email`, `many2many\_tags`, явное `many2one`, `radio`.

- `**editable**` (Режим редактирования дерева): `editable="bottom"` для встроенного редактирования One2many.

- `**options**` (Опции виджета): `{'no\_create': True}` для `many2one` в мастере отчета.

- `**groups**` (Ограничение по группам): использование групп на уровне полей в мастере отчета.

- `**class**` (Классы кнопок): `btn-primary`, `btn-secondary`, `btn-danger` в футерах форм мастеров.

- `**special**` (Специальное поведение кнопки): `special="cancel"` для закрытия диалогового окна.

- `**type**` (Тип графика): `type="pie"` в `<graph>`.

- `**compute**` (Вычисляемое поле, Python): используется в моделях, напр. `Transaction.name` (`@api.depends(...)`), а также `ResUsers.investor\_role\_type` с `compute` и `inverse`.

Замечание: атрибуты `domain` и `attrs` в XML данного модуля не применяются; фильтрация доменом реализована на уровне action-методов Python (`open\_account`, `open\_account` в `transaction.py`) для контекстного ограничения записей.

**Пример условного форматирования**

В модуле используется условное форматирование строк в списках транзакций.

- Форма: `view\_transaction\_tree`

- Атрибуты:

- `decoration-muted="operation\_type=='commission'"`

- `decoration-success="operation\_type=='deposit'"`

- `decoration-info="operation\_type in ['buy', 'sell']"`

- `decoration-danger="operation\_type=='withdrawal'"`

- Назначение: визуально выделять строки по типам операций; успешные — поступления (зачисления), информативные — торговые операции, опасные — списания, приглушенные — комиссии.

В других списках (`account`, `investor`, `asset`, `broker`) строковое условное форматирование не применяется.

**Вывод**

В ходе выполнения лабораторной работы произведен анализ и формализация пользовательских представлений модуля Odoo «Инвестиционный модуль». В модуле реализованы формы, списки, а также аналитические представления (Pivot и Graph) для сущности транзакций. Интерфейс организован с использованием иерархических форм и вкладок, обеспечивая структурированный ввод и просмотр данных: для счетов предусмотрено вложенное редактирование состава активов, а транзакции представлены как в табличном виде, так и в сводно-графическом. Реализованы вычисляемые поля и динамическая видимость элементов формы, а также условное форматирование строк списков транзакций. Разработанные представления соответствуют задачам предметной области и обеспечивают удобство эксплуатации для ролей «Инвестор», «Брокер» и «Администратор инвестиций».