Реферат

Содержание

[Реферат 3](#_Toc168306884)

[Содержание 4](#_Toc168306885)

[Введение 6](#_Toc168306886)

[1 Аналитический раздел 7](#_Toc168306887)

[1.1 Анализ предметной области 7](#_Toc168306888)

[1.2 Анализ существующих решений 8](#_Toc168306889)

[1.3 Формализация задачи 9](#_Toc168306890)

[1.4 Формализация данных 9](#_Toc168306891)

[1.5 Формализация ролей 11](#_Toc168306892)

[1.6 Анализ моделей баз данных 13](#_Toc168306893)

[Вывод из аналитического раздела 13](#_Toc168306894)

[2 Конструкторский раздел 14](#_Toc168306895)

[2.1 Сущности базы данных 14](#_Toc168306896)

[2.2 Роли базы данных 14](#_Toc168306897)

[2.3 Функции и процедуры базы данных 14](#_Toc168306898)

[2.4 Классы приложения 14](#_Toc168306899)

[Вывод из конструкторского раздела 14](#_Toc168306900)

[3 Технологический раздел 15](#_Toc168306901)

[3.1 Выбор СУБД 15](#_Toc168306902)

[3.2 Выбор средств реализации приложения 15](#_Toc168306903)

[3.3 Реализация сущностей и ограничений целостности 15](#_Toc168306904)

[3.4 Реализация функций и процедур 15](#_Toc168306905)

[3.5 Реализация ролевой модели 15](#_Toc168306906)

[3.6 Тестирование функций и процедур 15](#_Toc168306907)

[3.7 Интерфейс доступа к базе данных 15](#_Toc168306908)

[Вывод из технологического раздела 15](#_Toc168306909)

[4 Исследовательский раздел 16](#_Toc168306910)

[4.1 Описание задачи исследования 16](#_Toc168306911)

[4.2 Результаты 16](#_Toc168306912)

[Вывод из исследовательского раздела 16](#_Toc168306913)

[Заключение 17](#_Toc168306914)

[Список использованных источников 18](#_Toc168306915)

Введение

На сегодняшний день возможность анализировать и управлять личными финансами стала одним из основных требований успешного ведения домашнего хозяйства. В условиях быстро меняющейся экономической среды необходим удобный и простой инструмент, позволяющий отслеживать состояние банковских счетов. Одним из таких инструментов является приложение для сведения ежемесячного баланса доходов и расходов.

Во-первых, главная потребность в нем обусловлена растущим количеством и сложностью транзакций. Появилось не только огромное количество различных способов оплаты (наличный расчет, внутрибанковские переводы, эквайринг для физических и виртуальных карт, и т.д.), но и множество мелких расходов, что в совокупности приводит к невозможности быстро и точно определить, сколько было потрачено в той или иной категории.

Во-вторых, растущий темп жизни и повсеместная цифровизация общества требуют возможности своевременного онлайн-мониторинга своих финансов, при этом не затрачивающего большого количества времени.

Исходя из вышесказанного, можно сделать вывод о достаточной востребованности и актуальности подобного приложения на текущий момент.

Таким образом, целью данной курсовой работы является разработка программного обеспечения (приложения и базы данных) для анализа ежемесячных доходов и расходов. Для достижения поставленной цели необходимо решить следующие задачи:

1. проанализировать существующие решения;
2. сформулировать требования к разрабатываемому ПО и формализовать задачу;
3. проанализировать существующие базы данных и выбрать подходящие;
4. спроектировать базу данных, описать ее сущности, связи и процедуры;
5. реализовать приложение, позволяющее взаимодействовать с базой данных;
6. провести исследование быстродействия полученного приложения.

1 Аналитический раздел

В данном разделе будет проведен анализ предметной области, сравнительный анализ существующих аналогов, сформулированы требования к разрабатываемой базе данных и приложению, формализована и описана информация, подлежащая хранению в проектируемой базе данных и проведен анализ существующих баз данных на основе формализации данных.

На основе результатов вышеперечисленных действий также будут формализованы и описаны пользователи проектируемого приложения, а также построены ER-диаграмма сущностей в нотации Чена и диаграмма вариантов использования.

* 1. Анализ предметной области

Личный бюджет – персональный план доходов и расходов на определенный период – месяц, квартал, год [?]. Исходя из этого определения, можно считать процесс анализа ежемесячных транзакций процессом планирования бюджета. Тогда основной сущностью, лежащей в основе всей системы, очевидным образом становится транзакция – некоторая запись о факте перемещения денежных средств с одного банковского счета на другой. Также, в контексте анализа личных расходов и доходов кроме исходного счета, целевого счета и суммы, необходима дополнительная информация, включающая время транзакции (для отнесения ее к одному из периодических бюджетов), финансовую категорию, к которой принадлежит транзакция, а также дополнительную пользовательскую информацию.

Следующей важной сущностью является финансовая цель – некоторая цель, требующая наличия определенного количества свободных средств. В нашем случае будет достаточно привязать цель к определенному банковскому счету (т.н. «накопительный» счет) и позволить пользователю задать ожидаемое на нем значение.

Наконец, необходимо помнить о том, что личные накопления и траты могут включать несколько различных валют. Поэтому, для упрощения внесения информации о транзакциях между такими счетами, необходимо хранить также данные о используемых валютах и обменных курсах между ними.

* 1. Анализ существующих решений

Так как спрос на инструменты для анализа расходов и доходов не снижается на протяжении последнего десятилетия, на рынке уже существует множество готовых подходов и решений. Наиболее популярными и распространенными являются:

1. пользовательские реализации в Excel и других табличных процессорах;
2. встроенные аналитические модули в банковских приложениях;
3. Monefy;
4. Дзен-мани.

Сравнение данных решений (см. Таблица 1) будет проводится по следующим критериям:

1. возможность сохранять и анализировать информацию о счетах и транзакциях;
2. возможность запланировать бюджет;
3. возможность запланировать финансовые цели (например, накопление на покупку машины);
4. отсутствие жесткой привязки к платформе (десктоп, веб, мобильные устройства);
5. возможность анализировать транзакции без привязки к платежной системе;
6. возможность онлайн доступа;
7. бесплатность.

Таблица 1. Сравнение существующих решений

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Критерий | Excel и др. | Встроенные модули | Monefy | Дзен-мани |
| 1 | да | да | да | да |
| 2 | только вручную | не во всех | да | да |
| 3 | только вручную | не во всех | нет | да |
| 4 | нет | не во всех | нет | да |
| 5 | да | нет | да | да |
| 6 | нет | да | да | да |
| 7 | зависит от табличного процессора | не все | да | нет |

Ни одно из рассмотренных решений не удовлетворяет в полной мере выбранным критериям. Кроме того, стоит отметить, что на данный момент наиболее подходящее существующее решение (Дзен-мани) фактически находится в неработоспособном состоянии и не получало обновлений с 2017 года. Таким образом, реализуемое приложение выгодно отличается от существующих аналогов, удовлетворяя заявленным критериям.

* 1. Формализация задачи

Необходимо спроектировать базу данных для хранения информации о пользователях, банковских счетах, транзакциях, валютах, обменных курсах, транзакциях, финансовых категориях, бюджетах и целях.

Необходимо разработать приложение (веб-сервис), предоставляющее программный (в виде REST API) и пользовательский интерфейс для взаимодействия с базой данных, позволяющий просматривать, добавлять, изменять и удалять сущности, хранящиеся в базе данных, а также выполнять над ними дополнительные аналитические операции (подсчет готовности цели, расчет бюджета за период, поиск транзакций по категории).

Необходимо реализовать на стороне приложения механизм авторизации и предусмотреть наличие трех ролей: пользователь, администратор и система.

* 1. Формализация данных

На основе приведенного выше анализа предметной области выделены следующие категории данных:

1. пользователь;
2. банковский счет;
3. финансовая категория;
4. валюта;
5. обменный курс;
6. бюджет;
7. финансовая цель;
8. транзакция.

Подробная информация о каждой категории приведена в таблице 2.

Таблица 2. Категории данных

|  |  |
| --- | --- |
| Категория | Информация |
| Пользователь | Id, логин, хэш пароля, флаг администраторского доступа |
| Банковский счет | Id, id валюты, описание, текущее значение |
| Финансовая категория | Id, название категории |
| Валюта | Id, название валюты |
| Обменный курс | Id, id исходной валюты, id целевой валюты, коэффициент обмена |
| Бюджет | Id, id валюты, начало периода, конец периода, описание, начальный объем средств |
| Финансовая цель | Id, id банковского счета, описание, целевое значение счета |
| Транзакция | Id, id финансовой категории, id исходного счета, id целевого счета, значение транзакции, описание, временная метка |

На рисунке 1 приводится ER-диаграмма сущностей в нотации Чена.

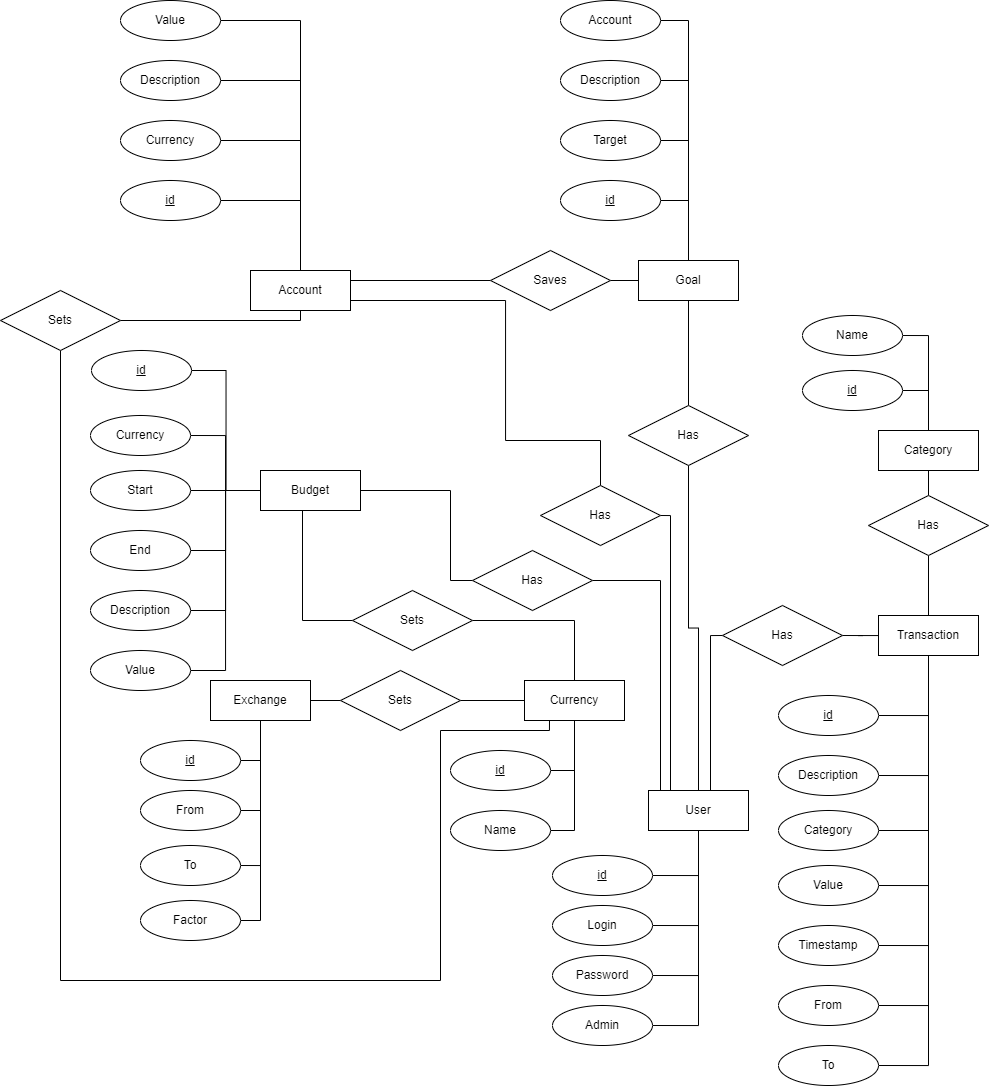


Рисунок 1. ER-диаграмма в нотации Чена

* 1. Формализация пользователей

Выделим категории пользователей, используемые для доступа к приложению:

1. пользователь
2. администратор
3. система (сервис).

Пользователь обладает базовым уровнем доступа, позволяющим вести бюджет: может просматривать валюты, категории и обменные курсы; просматривать и редактировать свою учетную запись; просматривать, создавать, изменять и удалять бюджеты, счета, цели и транзакции.

Администратор обладает всеми возможностями пользователя и дополнительно имеет доступ к созданию, изменению и удалению категорий, валют и обменных курсов. Категория существует для возможности поддержания системы в актуальном состоянии.

И, наконец, система обладает полным доступом к базе данных, может управлять существующими и создавать новые таблицы, создавать и удалять пользователей. Выполняет техническое обслуживания всей системы и регистрацию учетных записей пользователей.

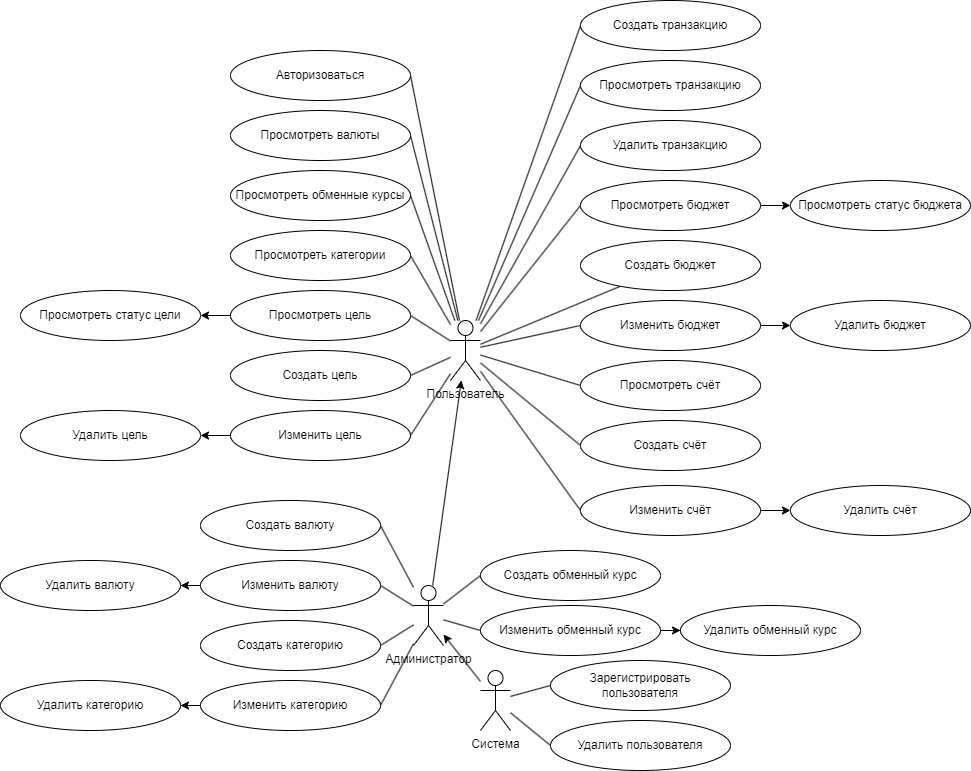
На рисунке 2 для описанных категорий приведена диаграмма вариантов использования.

Рисунок 2. Диаграмма вариантов использования

* 1. Анализ моделей баз данных

Термин «база данных» впервые появился более шестидесяти лет назад, и за это время получил большое количество интерпретаций, в том числе и от многих авторитетных авторов. Несмотря на это, общепризнанная единая формулировка отсутствует. Поэтому, учитывая специфику данной работы и избегая отдавать предпочтение какому-либо стороннему источнику, воспользуемся определением из курса лекций Ю. М. Гавриловой.

База данных – совокупность данных, организованных по определенным правилам, предусматривающая общие принципы описания, хранения и манипулирования, независимая от прикладных программ [?].

Для обеспечения возможности взаимодействия с базой данных используется система управления базами данных (СУБД) – программный компонент, отвечающий за обработку всех пользовательских запросов, составленных на формальном языке, абстрагированным от деталей реализации.

Одним из основных признаков, использующихся для классификации СУБД, является модель данных – абстрактное, самодостаточное, логическое определение объектов, операторов и прочих элементов, составляющих абстрактную машину, с которой взаимодействует пользователь [?].

Существуют следующие основные модели данных:

1. дореляционные;
2. реляционные;
3. постреляционные.
   * 1. Дореляционные модели

Данный тип включает модель на основе инвертированных списков, иерархические и сетевые модели.

Базы данных, организованные с помощью инвертированных списков, напоминают реляционные с тем отличием, что хранимые таблицы и пути доступа к ним видны пользователям [?].

Иерархические базы данных состоят из упорядоченного набора нескольких экземпляров одного типа дерева. Хранимые записи поддерживают связь «предок-потомок», в которой предок может иметь несколько потомков, но потомок может иметь только одного предка [?]. Такой подход не только создает определенные трудности с реализацией различных моделей отношений между сущностями, но и зачастую приводит к массовым изменениям при вставке или удалении.

Сетевая модель является расширением иерархической, позволяя потомку иметь несколько предков.

Главным недостатком дореляционных моделей, полностью нивелирующим их плюсы (эффективное использование памяти и пространство для оптимизаций), является жесткая зависимость от физической организации базы данных, приводящая к высокой сложности внесения каких-либо изменений.

* + 1. Реляционные модели

Реляционная модель обеспечивает представление данных в виде двумерной таблицы-отношения, состоящей из набора столбцов и строк [?]. Столбцы содержат информацию об атрибутах отношения – имя и тип; строки, отличные от заглавной строки, фактически являются кортежами, содержащими один компонент для каждого атрибута [?].

Главными плюсами данного подхода, обеспечившими ему широкую популярность, являются абстрагированность от физической реализации и простота поддержания целостности благодаря хранению информации о типах атрибутов.

Недостатки же включают требование атомарности для атрибутов и трудности с эффективным использованием памяти в отличие от дореляционных моделей.

* + 1. Постреляционные модели

Постреляционные модели представляют собой расширение реляционных, снимая ограничение на неатомарные атрибуты. Это делает данные модели более удобными для хранения составных сущностей, описанных с применением объектно-ориентированного подхода, и значительно повышает эффективность обработки в тех случаях, когда поле сущности является некой последовательностью данных (массив, список, множество и т.д.).

Основным недостатком модели является возросшая сложность поддержания целостности составных атрибутов.

Вывод из аналитического раздела

В данном разделе был проведен анализ предметной области, сравнительный анализ существующих аналогов, в результате которого ни один из них не удовлетворил в полной мере выбранным критериям.

На основе анализа предметной области были сформулированы требования к разрабатываемой базе данных и приложению, формализована и описана информация, подлежащая хранению в проектируемой базе данных и проведен анализ существующих баз данных на основе формализации данных.

По итогам анализа была выбрана реляционная модель, как наиболее соответствующая сформулированным требованиям (нет составных атрибутов, необходимо поддерживать целостность).

Также были формализованы и описаны пользователи проектируемого приложения и построены ER-диаграмма сущностей в нотации Чена и диаграмма вариантов использования.

1. Конструкторский раздел

В данном разделе будут описаны сущности проектируемой базы данных, ограничения целостности, проектируемые процедуры и функции в формате схемы и ролевая модель на уровне базы данных, а также приведена ER-диаграмма проектируемой базы данных.

* 1. Сущности базы данных

На основе результатов формализации категорий данных, приведенных в таблице 2, выделяются следующие таблицы:

1. Currencies – валюты;
2. Exchanges – обменные курсы валют;
3. Categories – финансовые категории;
4. Users – пользователи;
5. Budgets – планируемые бюджеты;
6. Accounts – банковские счета;
7. Goals – финансовые цели;
8. Transactions – транзакции.

Подробная информация для каждой сущности о структуре и типах столбцов, а также ограничениях целостности приведена в таблицах 3, 4, 5, 6, 7, 8, 9, 10.

Таблица 3. Currencies

|  |  |  |  |
| --- | --- | --- | --- |
| Атрибут | Тип | Ограничения | Комментарий |
| id | serial | primary key, not null | Идентификатор валюты |
| name | text | not null | Название валюты |

Таблица 4. Exchanges

|  |  |  |  |
| --- | --- | --- | --- |
| Атрибут | Тип | Ограничения | Комментарий |
| id | serial | primary key, not null | Идентификатор обменного курса |
| from | int | foreign key, not null | Исходная валюта |
| to | int | foreign key, not null | Целевая валюта |
| factor | double precision | not null | Обменный коэффициент |

Таблица 5. Categories

|  |  |  |  |
| --- | --- | --- | --- |
| Атрибут | Тип | Ограничения | Комментарий |
| id | serial | primary key, not null | Идентификатор категории |
| name | text | not null | Название категории |

Таблица 6. Users

|  |  |  |  |
| --- | --- | --- | --- |
| Атрибут | Тип | Ограничения | Комментарий |
| id | serial | primary key, not null | Идентификатор пользователя |
| login | text | not null | Логин пользователя |
| password | text | not null | Хэш пароля |
| admin | bool | not null | Флаг прав администратора |

Таблица 7. Budgets

|  |  |  |  |
| --- | --- | --- | --- |
| Атрибут | Тип | Ограничения | Комментарий |
| id | serial | primary key, not null | Идентификатор бюджета |
| owner | int | foreign key, not null | Пользователь, создавший бюджет |
| currency | int | foreign key, not null | Валюта бюджета |
| start | timestamp | not null | Начало бюджетного периода |
| end | timestamp | not null | Конец бюджетного периода |
| description | text | not null | Описание бюджета |
| value | double precision | not null | Начальное значение бюджета (количество средств, доступное с начала периода) |

Таблица 8. Accounts

|  |  |  |  |
| --- | --- | --- | --- |
| Атрибут | Тип | Ограничения | Комментарий |
| id | serial | primary key, not null | Идентификатор счета |
| owner | int | foreign key, not null | Пользователь, создавший счет |
| currency | int | foreign key, not null | Валюта счета |
| description | text | not null | Описание счета |
| value | double precision | not null | Текущее значение счета |

Таблица 9. Goals

|  |  |  |  |
| --- | --- | --- | --- |
| Атрибут | Тип | Ограничения | Комментарий |
| id | serial | primary key, not null | Идентификатор цели |
| owner | int | foreign key, not null | Пользователь, создавший цель |
| account | int | foreign key, not null | Счет, к которому привязана цель |
| description | text | not null | Описание цели |
| target | double precision | not null | Сумма, которую необходимо достигнуть на счете |

Таблица 10. Transactions

|  |  |  |  |
| --- | --- | --- | --- |
| Атрибут | Тип | Ограничения | Комментарий |
| id | serial | primary key, not null | Идентификатор транзакции |
| owner | int | foreign key, not null | Пользователь, создавший транзакцию |
| category | int | foreign key, not null | Категория транзакции |
| from | int | foreign key | Исходный счет (null при внешнем поступлении) |
| to | int | foreign key | Целевой счет (null при внешнем переводе) |
| value | double precision | not null | Значение |
| description | text | not null | Описание транзакции |
| timestamp | timestamp | not null | Временная отметка |

Тип serial здесь представляет авто-инкрементируемую версию типа int.

Невозможность создать транзакцию, у которой поля from и to одновременно будут иметь значение null, контролируется приложением.

На рисунке 3 приведена ER-диаграмма проектируемой базы данных.

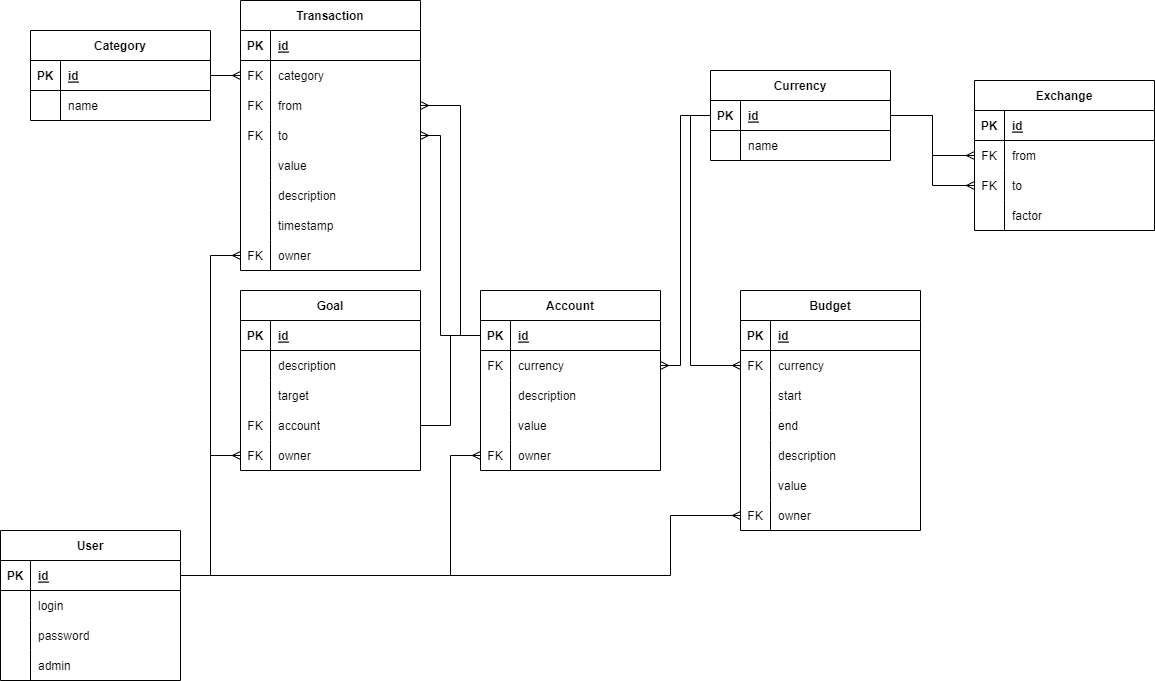


Рисунок 3. ER-диаграмма базы данных

* 1. Роли базы данных

На основе результатов формализации пользователей базы данных, проведённой в аналитическом разделе, определена ролевая модель, включающая три роли:

1. user – имеет право доступа select для таблиц currencies, exchanges, categories, users; имеет право доступа update для таблицы users и имеет полные права доступа к остальным таблицам (не может добавлять и удалять пользователей и изменять общедоступные сущности);
2. admin – имеет права доступа select и update для таблицы users; имеет полные права доступа к остальным таблицам (имеет доступ ко всем действиям, кроме добавления и удаления пользователей);
3. service – имеет полные права доступа ко всем таблицам и базе данных.
   1. Функции и процедуры базы данных

Для удобства работы с базой данных, а также корректной организации вычислений бюджета и целей спроектированы нижеописанные функции и процедуры.

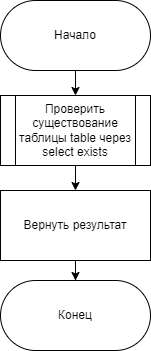
1. check\_table\_exists (table) – функция, проверяющая существование таблицы table. В случае успеха возвращает true, иначе false. Схема приведена на рисунке 4.
2. check\_tables\_exist (tables) – функция, проверяющая существование всех таблиц из массива tables. Если все таблицы существуют, возвращает true, иначе false. Схема приведена на рисунке 5.
3. convert\_value (from, to, value) – функция, преобразующая значение из валюты from в валюту to. Если валюты равны, значение не изменяется. Схема приведена на рисунке 6.
4. get\_budget\_status (id, owner) – функция, вычисляющая статистику для бюджета с идентификатором id, принадлежащего пользователю owner. Определяет, сколько средств за планируемый период было потрачено и получено, а также считает общую сумму. Не учитывает внутренние транзакции (переводы между счетами, принадлежащими пользователю). Схема приведена на рисунке 7.
5. get\_goal\_status (id) – функция, вычисляющая статистику для финансовой цели с идентификатором id. Определяет, сколько процентов цели выполнено, сколько средств накоплено и сколько осталось накопить. Схема приведена на рисунке 8.
6. add\_account\_value (id, currency, value) – процедура, изменяющая значение счёта с идентификатором id на значение value в валюте currency. Если указанная валюта не соответствует валюте счета, функция дополнительно произведёт конвертацию. Схема приведена на рисунке 9.
7. add\_transaction (owner, category, from, to, value, description, timestamp) – функция, выполняющая добавление транзакции с указанными атрибутами и применение её к содержимому счетов from и to. Если транзакция внешняя, её сумма будет считаться указанной в валюте счета, иначе будет произведена конвертация. Функция возвращает идентификатор созданной транзакции. Схема приведена на рисунке 10.
8. del\_transaction (id) – процедура, выполняющая удаление транзакции с идентификатором id и её откат для задействованных счетов. Выполняет преобразования, обратные тем, что делает функция add\_transaction. Схема приведена на рисунке 11.

Рисунок 4. Функция check\_table\_exists

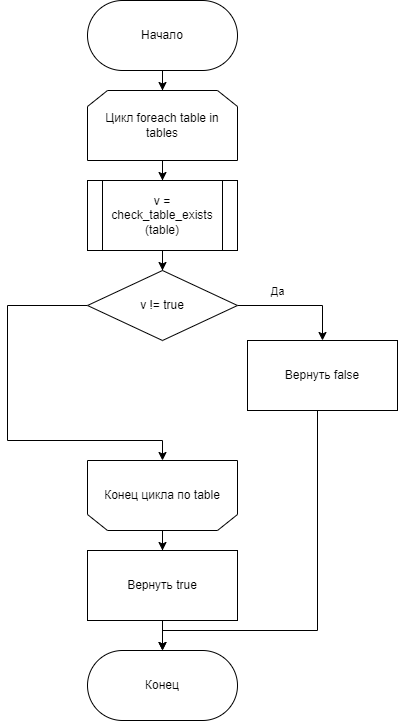
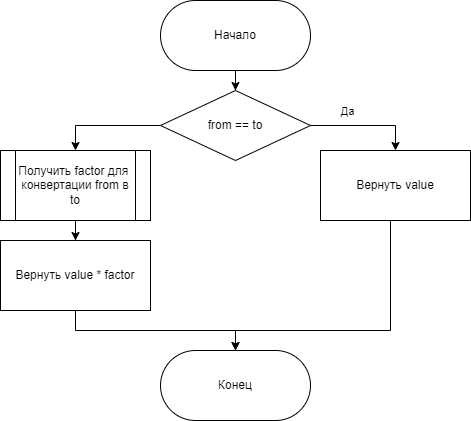


Рисунок 5. Функция check\_tables\_exist

Рисунок 6. Функция convert\_value

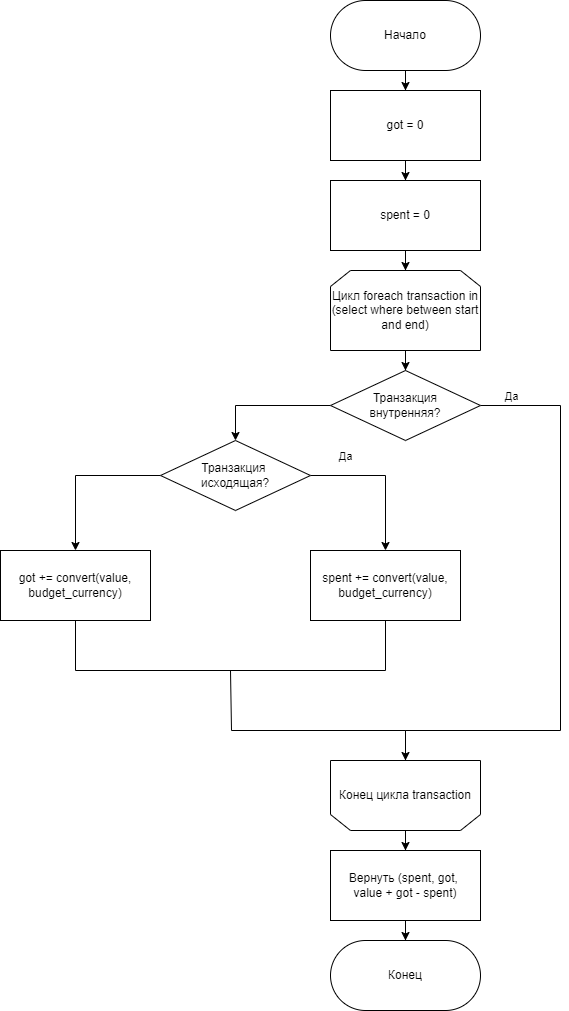
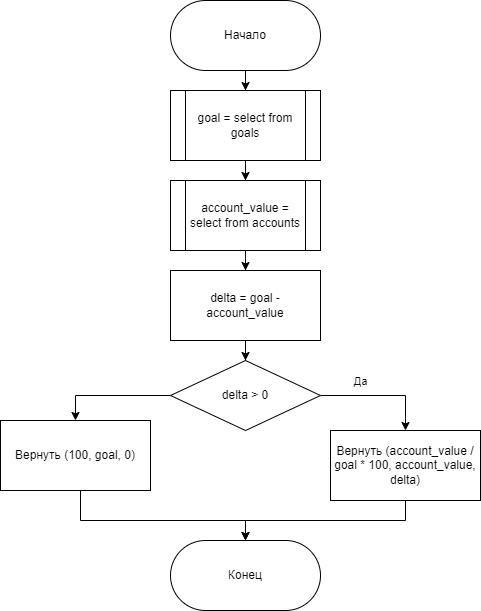


Рисунок 7. Функция get\_budget\_status

Рисунок 8. Функция get\_goal\_status

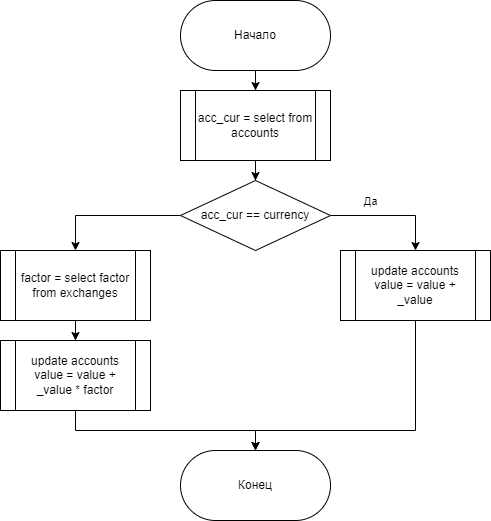


Рисунок 9. Процедура add\_account\_value

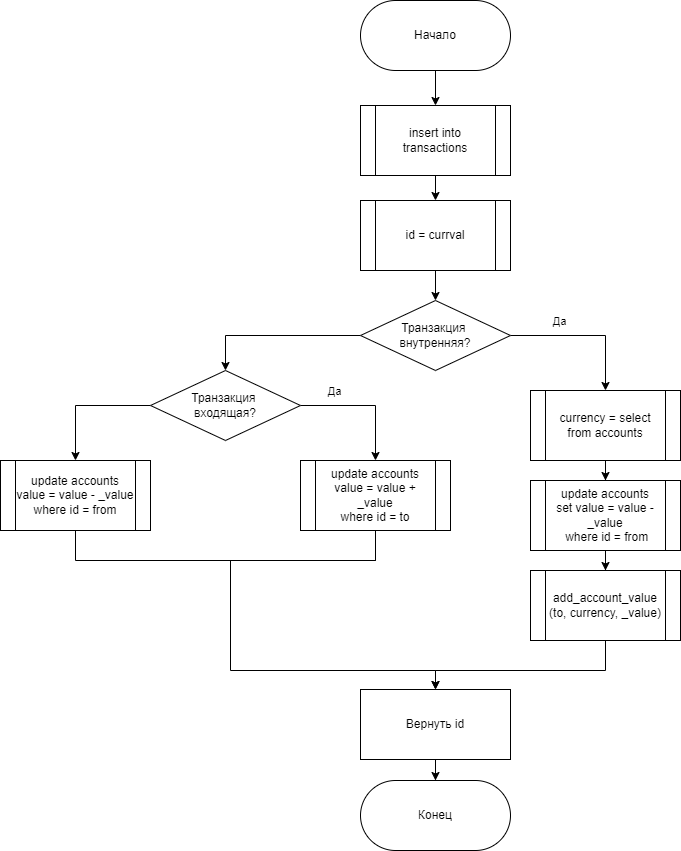
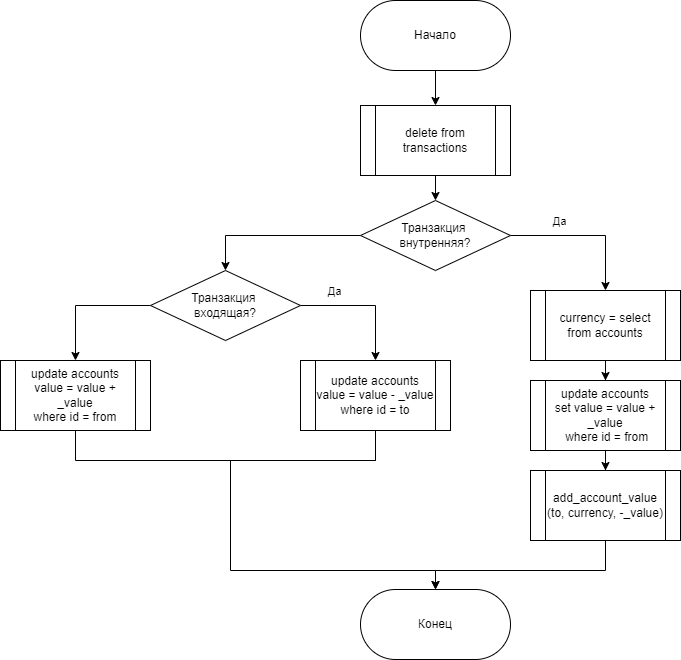
Рисунок 10. Функция add\_transaction

Рисунок 11. Процедура del\_transaction

Вывод из конструкторского раздела

В данном разделе были описаны сущности проектируемой базы данных, прокомментированы их атрибуты и приведены ограничения целостности.

Были перечислены проектируемые процедуры и функции, описаны и представлены в формате блок-схемы.

Была описана ролевая модель на уровне базы данных.

Также была приведена ER-диаграмма проектируемой базы данных.

1. Технологический раздел

В данном разделе будет рассмотрен выбор средств реализации базы данных и приложения, описаны сущности реализованной базы данных, описаны реализованные ограничения целостности базы данных, приведена реализация ролевой модели, приведены все реализованные процедуры и функции, описаны методы тестирования и тесты для всех разработанных на стороне базы функций. Также будет описан интерфейс доступа к базе данных.

3.1 Выбор СУБД

3.2 Выбор средств реализации приложения

3.3 Реализация сущностей и ограничений целостности

3.4 Реализация функций и процедур

3.5 Реализация ролевой модели

3.6 Тестирование функций и процедур

3.7 Интерфейс доступа к базе данных

Вывод из технологического раздела

4 Исследовательский раздел

4.1 Описание задачи исследования

4.2 Результаты

Вывод из исследовательского раздела

Заключение

Список использованных источников

1. Погорелов Д. А. [и др.] Булевы операции на трехмерных моделях в компьютерной графе [Электронный ресурс] – Режим доступа: <https://alley-science.ru/domains_data/files/january_1/BULEVY%20OPERACII%20NA%20TRYoHMERNYH%20MODELYaH%20V%20KOMPYuTERNOY%20GRAFIKE.pdf> (дата обращения: 16.09.2023)
2. Carr D. Computation of Potentially Visible Set for Occluded Three-Dimensional Environments [Электронный ресурс] – Режим доступа: <https://www.bc.edu/content/dam/files/schools/cas_sites/cs/pdf/academics/honors/04DerekCarr.pdf> (дата обращения: 20.10.2023)
3. Роджерс Д. Алгоритмические основы машинной графики [Текст]. Пер. с англ. С.А.Вичеса, Г.В. Олохтоновой, П.А. Монахова. – М.: Мир, 1989. – 512 с.
4. OpenGL Perspective [Электронный ресурс] – Режим доступа: <https://registry.khronos.org/OpenGL-Refpages/gl2.1/xhtml/gluPerspective.xml> (дата обращения: 20.10.2023)
5. OpenGL LookAt [Электронный ресурс] – Режим доступа: <https://registry.khronos.org/OpenGL-Refpages/gl2.1/xhtml/gluLookAt.xml> (дата обращения 20.10.2023)
6. The Java Language Specification [Электронный ресурс] – Режим доступа: <https://docs.oracle.com/javase/specs/jls/se11/html/index.html> (дата обращения 14.11.2023)
7. IntelliJ IDEA [Электронный ресурс] – Режим доступа: <https://www.jetbrains.com/ru-ru/idea/> (дата обращения 14.11.2023)
8. Java SE 11 [Электронный ресурс] – Режим доступа: <https://www.oracle.com/java/technologies/javase/jdk11-archive-downloads.html> (дата обращения 14.11.2023)
9. Gradle Build Tool [Электронный ресурс] – Режим доступа: <https://gradle.org> (дата обращения 14.11.2023)
10. JMH [Электронный ресурс] – Режим доступа: <https://openjdk.org/projects/code-tools/jmh/> (дата обращения 17.11.2023)