

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	2
1 Аналитическая часть	3
1.1 Анализ предметной области	3
1.2 Анализ существующих решений и требования к разрабатываемым базе данных и приложению	5
1.3 Системы управления базами данных	6
1.4 Описание сущностей проектируемой базы данных	7
1.5 Описание пользователей проектируемого приложения	8
2 Конструкторская часть	10
2.1 Формализация бизнес-правил	10
2.2 Проектирование базы данных	19
2.3 Ролевая модель проектируемой базы данных	23
2.4 Триггеры	24
2.5 Структура разрабатываемого приложения	29
3 Технологическая часть	31
3.1 Выбор средств реализации	31
3.2 Описание сущностей и ограничений целостности базы данных	33
3.3 Реализация триггеров	35
3.4 Описание ролевой модели на уровне базы данных	38
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	40

ВВЕДЕНИЕ

Целью данной курсовой работы является разработка базы данных сыгранных на кубке мира шахматных партий.

Задачи работы:

- проведение анализа предметной области, связанной с шахматными турнирами;
- формулировка требований к базе данных и приложению;
- описание пользователей проектируемого приложения;
- проектирование архитектуры базы данных и ограничений целостности;
- проектирование ролевой модели на уровне базы данных;
- анализ и выбор средств реализации базы данных и приложения;
- реализация спроектированной базы данных и необходимого интерфейса для взаимодействия с ней;
- исследование характеристик разработанного программного обеспечения.

1 Аналитическая часть

1.1 Анализ предметной области

Проведение кубка мира по шахматам регламентируется международной шахматной федерацией (ФИДЕ) [1]. В турнире принимают участие 206 человек [1]. Соревнование состоит из восьми раундов [1]. Количество участников в каждом раунде представлено в таблице 1.1.

Таблица 1.1 — Количество участников в каждом раунде кубка мира по шахматам

Раунд	Количество участников
Раунд 1	156
Раунд 2	128 (78 победителей раунда 1 и 50 игроков с самым высоким рейтингом)
Раунд 3	64
Раунд 4	32
Раунд 5	16
Раунд 6	8
Раунд 7	4
Раунд 8, матч за третье место	2
Раунд 8, финал	2

Турнир проводится по нокаут-системе [1]. В каждом раунде матч между двумя шахматистами состоит из двух партий с временным контролем ФИДЕ: участникам дается 90 минут на первые 40 ходов, затем — 30 минут на оставшуюся часть игры с увеличением времени на 30 секунд после каждого хода, начиная с первого хода [1]. Игрок, набравший по прошествии двух партий большее число очков, чем его соперник, становится победителем матча и переходит на следующий раунд [1].

Если количество очков игроков матча одинаковы, проводится тай-брейк: партии разыгрываются в соответствии с таблицей 1.2 до тех пор, пока не будет выявлен победитель [1].

Таблица 1.2 — Проведение тай-брейков

Номер тай-брейка	Количество партий	Время на каждого игрок, мин	Увеличение времени, сек	Примечания
1	2	25	10	—
2	2	10	10	—
3	2	5	3	—
4	1	3	2	—
5	1	3	2	Предварительная смена цветов фигур
6 и далее	1	3	2	—

После каждой сыгранной партии меняется рейтинг шахматистов, который формируется международной шахматной федерацией на основе метода расчета Эло [2, 3]. Перед вычислением нового рейтинга определяется вероятность PD достижения игроком результата в каждой партии по таблице 1.3.

Таблица 1.3 — Таблица преобразования разницы в рейтинге D в вероятность достижения результата PD игроком с более высоким рейтингом H и игроком с более низким рейтингом L , соответственно

D	PD		D	PD		D	PD		D	PD	
	H	L		H	L		H	L		H	L
0–3	.50	.50	92–98	.63	.37	198–206	.76	.24	345–357	.89	.11
4–10	.51	.49	99–106	.64	.36	207–215	.77	.23	358–374	.90	.10
11–17	.52	.48	107–113	.65	.35	216–225	.78	.22	375–391	.91	.09
18–25	.53	.47	114–121	.66	.34	226–235	.79	.21	392–411	.92	.08
26–32	.54	.46	122–129	.67	.33	236–245	.80	.20	412–432	.93	.07
33–39	.55	.45	130–137	.68	.32	246–256	.81	.19	433–456	.94	.06
40–46	.56	.44	138–145	.69	.31	257–267	.82	.18	457–484	.95	.05
47–53	.57	.43	146–153	.70	.30	268–278	.83	.17	485–517	.96	.04
54–61	.58	.42	154–162	.71	.29	279–290	.84	.16	518–559	.97	.03
62–68	.59	.41	163–170	.72	.28	291–302	.85	.15	560–619	.98	.02
69–76	.60	.40	171–179	.73	.27	303–315	.86	.14	620–735	.99	.01
77–83	.61	.39	180–188	.74	.26	316–328	.87	.13	>735	1.0	.00
84–91	.62	.38	189–197	.75	.25	329–344	.88	.12			

После получения значения PD для конкретного игрока вычисляется величина:

$$\Delta R = Res - PD, \quad (1.1)$$

где Res — результат партии (1 — выигрыш, 0.5 — ничья, 0 — проигрыш) [2].

Изменение рейтинга игрока вычисляется по формуле:

$$\Delta S = \Sigma \Delta R \cdot K, \quad (1.2)$$

где $\Sigma \Delta R$ — сумма всех ΔR для турнира или рейтингового периода,

$K = 20$, если рейтинг игрока не превышает 2400,

$K = 10$, если рейтинг игрока превышает 2400 [2].

Если количество партий n для игрока в любом списке за рейтинговый период, умноженное на K (как определено выше), превышает 700, то в качестве K выбирается такое наибольшее целое число, что произведение числа K на n не превышает 700 [2].

Во время проведения кубка мира по шахматам возможен прием ставок — подкрепленных деньгами прогнозов на исходы каких-либо событий [4]. Для каждого условия устанавливается коэффициент, определяющий чистую прибыль участников при выигрыше спора [4].

Существуют следующие виды ставок [4]:

- главная линия (исход партии — победа первого участника, ничья и т.д.),
- тоталы (количество свершенных событий в течение партии, матча, раунда или турнира),
- форы (с каким отрывом выиграет участник),
- экспрессы (объединение нескольких исходов в одну ставку),
- системы (объединение экспрессов в одну ставку).

В шахматных турнирах форы быть не может, поскольку во время партии не ведется счет очков, поэтому вводится понятие нулевой форы, при которой ставка делается на том, что один из участников, как минимум, не проиграет [5].

При объединении исходов в экспресс их коэффициенты перемножаются, при этом для выигрыша необходимо, чтобы все поставленные условия были выполнены [4].

Системы имеют размерность — пару натуральных чисел, определяющих количество исходов, из которых составляются экспрессы, и количество событий в каждом из экспрессов [4]. Выигранная денежная сумма вычисляется на основе коэффициентов свершенных экспрессов [4].

1.2 Анализ существующих решений и требования к разрабатываемым базе данных и приложению

В таблице 1.4 представлен сравнительный анализ существующих аналогов базы данных для хранения информации о сыгранных шахматных партиях.

Таблица 1.4 — Сравнительный анализ существующих аналогов

Название	Тип программного обеспечения	Открытый исходный код	Бесплатный	Возможность делать ставки
Scid [6]	Десктопное	+	+	-
ChessGames [7]	Веб-сайт	-	+	-
365Chess [8]	Веб-сайт	-	+	-
FicsGames [9]	Веб-сайт	-	+	-

Требования к разрабатываемым базе данных и приложению:

- база данных должна быть разработана для хранения информации о сыгранных в кубке мира шахматных партиях, выполненных в них ходах, игроках, судьях, ставках и пользователях системы;
- приложение должно быть разработано для работы с информацией, хранящейся в базе данных;
- пользователю должна быть предоставлена возможность добавлять, удалять и получать информацию о сыгранных шахматных партиях, выполненных в них ходах, игроках, судьях, ставках и пользователях системы;
- в приложении должно быть предусмотрено логирование на всех уровнях;
- приложение должно иметь два кэша: для передачи запросов от приложения в систему управления базой данных и для передачи информации от бэкенда во фронтенд.

1.3 Системы управления базами данных

Система управления базой данных (СУБД) — программное обеспечение, предоставляющее создание, обновление, хранение и поиск информации в базе данных [10]. СУБД имеет следующие функции [10]:

- предоставление средств определения данных в виде исходной формы и преобразование этих определений в соответствующую объектную форму;
- обработка запросов пользователя на выборку, изменение, добавление или удаление данных;
- оптимизация способа выполнения каждого из запросов;
- защита и поддержка целостности данных;
- восстановление данных;
- поддержка параллельности;
- выполнение функций с максимально возможной эффективностью.

По направленности системы управления базами данных классифицируются следую-

щим образом [11]:

- универсальные,
- специализированные.

Универсальные СУБД не ориентированы на какую-либо предметную область [11]. Каждая система такого рода является универсальной и реализует функционально избыточное множество операций над данными [11].

Специализированные СУБД предназначены для конкретных классов задач, которые не могут быть оптимально решены с помощью универсальных систем [11].

По способу доступа системы управления базами данных делятся на [11]:

- клиент-серверные,
- файл-серверные,
- встраиваемые.

В файл-серверных СУБД база данных хранится на специализированном сервере, а СУБД запускается на стороне клиента [11]. Как правило, такая архитектура используется внутри локальной сети [11]. Многопользовательская синхронизация осуществляется на уровне сервера, что влечёт многочисленные блокировки файлов [11]. Сервер только хранит данные, и не обрабатывает их, поэтому ответ от такого файл-сервера будет в виде блока данных для каждого запроса, что влечет за собой высокую нагрузку на сеть при большом числе клиентов [11]. Файл-серверные СУБД считаются устаревшими и пригодны для работы при малом количестве клиентов (до нескольких десятков) [11].

В клиент-серверных СУБД все основные компоненты выполняются на отдельном сервере, содержащем базу данных [11]. На клиенте находится интерфейсная часть СУБД и выполняется код приложения [11]. Достоинством клиент-серверной архитектуры является оптимизация на стороне клиента: всю работу с базой данных осуществляет сервер, по сети передаётся только обработанный ответ [11]. Недостатком клиент-серверной архитектуры является зависимость клиентов от сервера, так как, в случае неисправной работы последнего, теряется возможность получения доступа к базе данных [11].

Встраиваемые СУБД подразумевают хранение базы данных в оперативной памяти клиента во время работы приложения [11]. Как правило, встраиваемые СУБД представляют собой библиотеки, функции которой используются в коде программы [11]. Основным недостатком встраиваемых СУБД является незащищенность базы данных от программы клиента [11].

1.4 Описание сущностей проектируемой базы данных

На рисунке 1.1 представлена ER-диаграмма сущностей предметной области в нотации Чена.

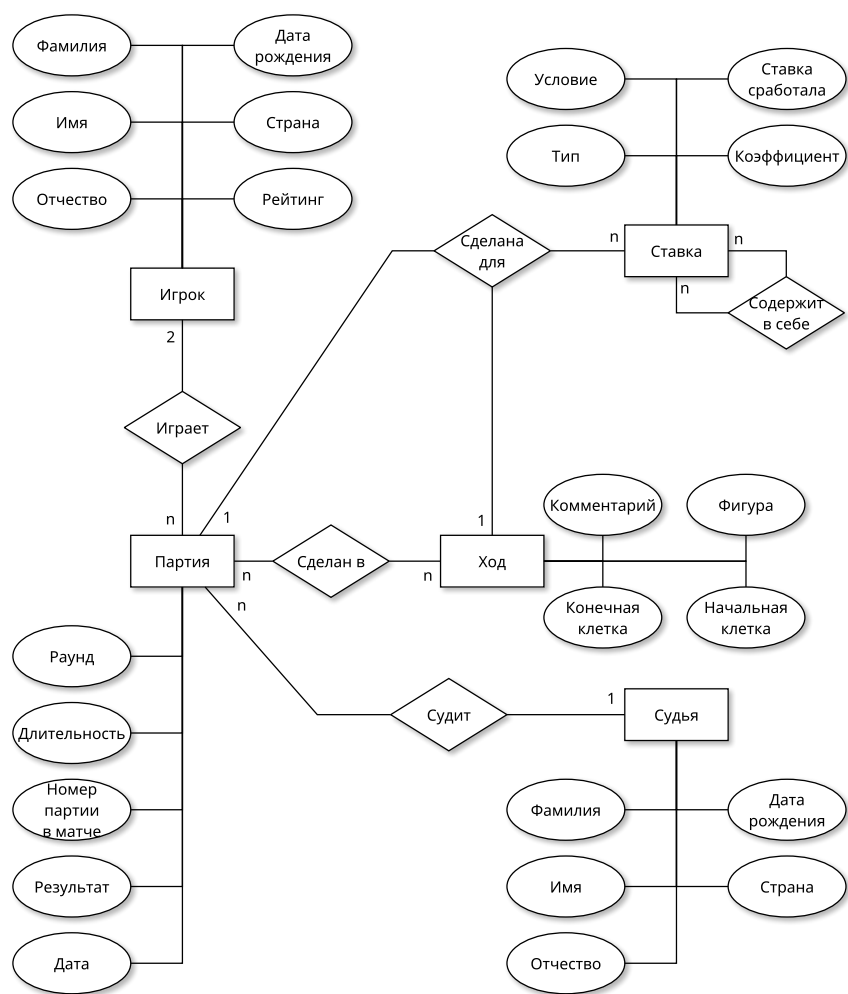


Рисунок 1.1 — ER-диаграмма сущностей предметной области

1.5 Описание пользователей проектируемого приложения

На рисунке 1.2 представлена диаграмма вариантов использования проектируемого приложения.

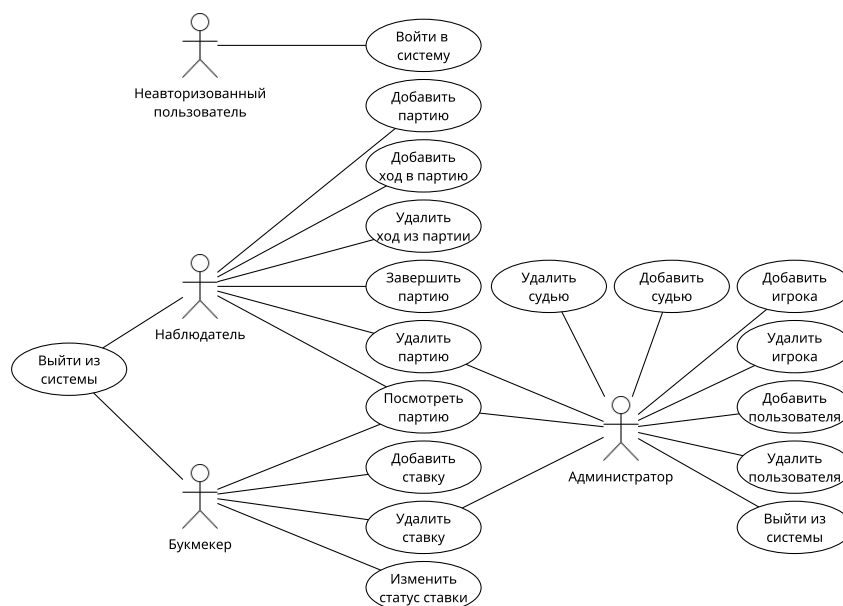


Рисунок 1.2 — Диаграмма вариантов использования проектируемого приложения

Неавторизованный пользователь не имеет доступа к информации, хранящейся в базе данных, поэтому для получения соответствующих прав ему необходимо войти в систему.

Наблюдатель — человек, ответственный за ввод шахматных партий в базу данных. Наблюдателю дана возможность добавлять и удалять партии и ходы, выполненные в них.

Букмекер — лицо, осуществляющее прием ставок от клиентов, желающих заключить пари. Букмекер имеет возможность добавлять и удалять информацию о спорах.

Администратор — лицо, ответственное за добавление и удаление пользователей системы и игроков из базы данных. Помимо основных функций, администратор может удалять информацию о шахматных партиях и ставках.

Вывод

В аналитическом разделе был проведен анализ предметной области, связанной с проведением кубка мира по шахматам и ставками на спорт. Были рассмотрены и сравнены существующие решения для хранения шахматных партий. Были сформулированы требования к проектируемым программному обеспечению и базе данных. Были рассмотрены системы управления базами данных на основе формализованной задачи. Были описаны сущности проектируемой базы данных и пользователи разрабатываемого приложения.

2 Конструкторская часть

2.1 Формализация бизнес-правил

На рисунках 2.1–2.18 представлены бизнес-правила проектируемого приложения.

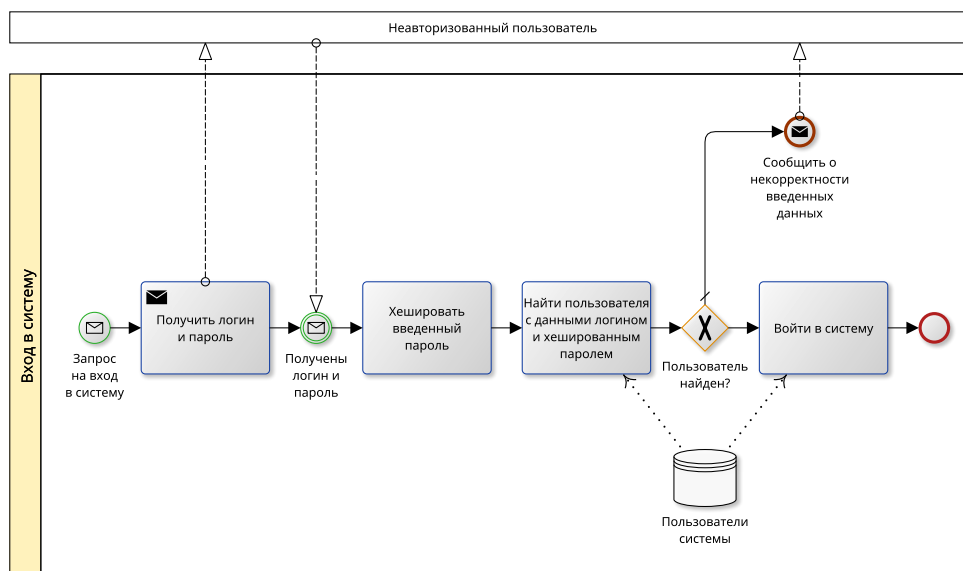


Рисунок 2.1 — Бизнес-правило входа в систему

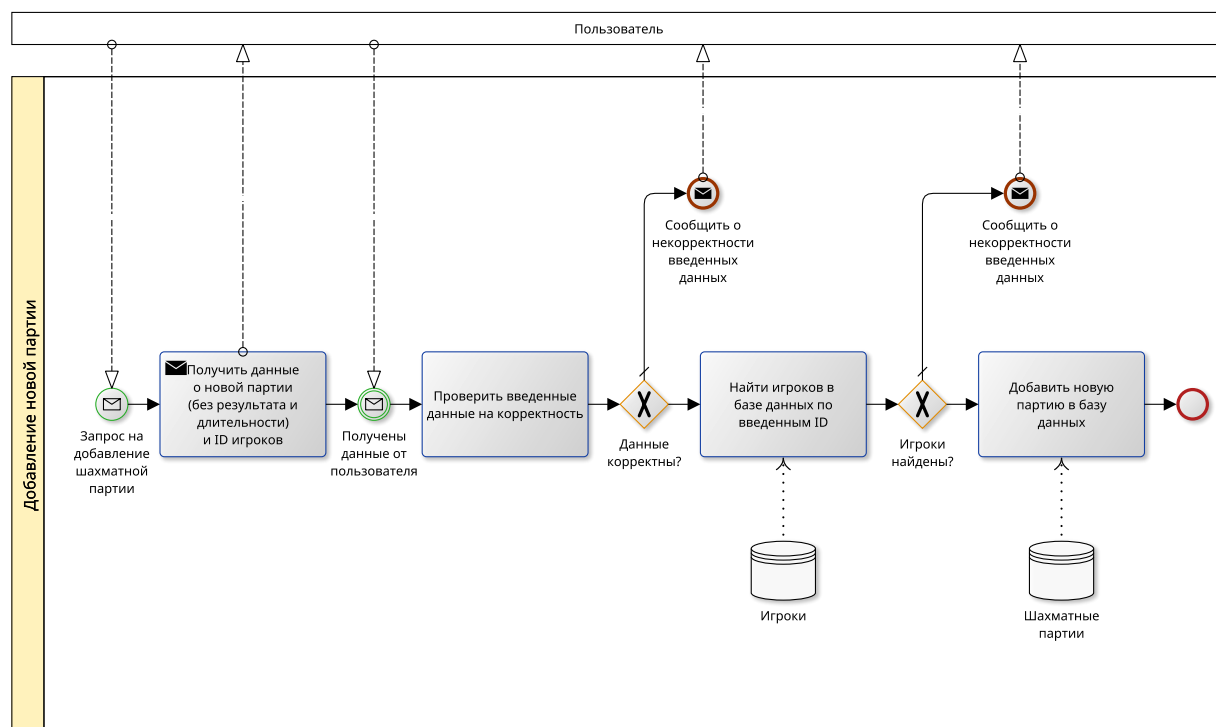


Рисунок 2.2 — Бизнес-правило добавления новой шахматной партии

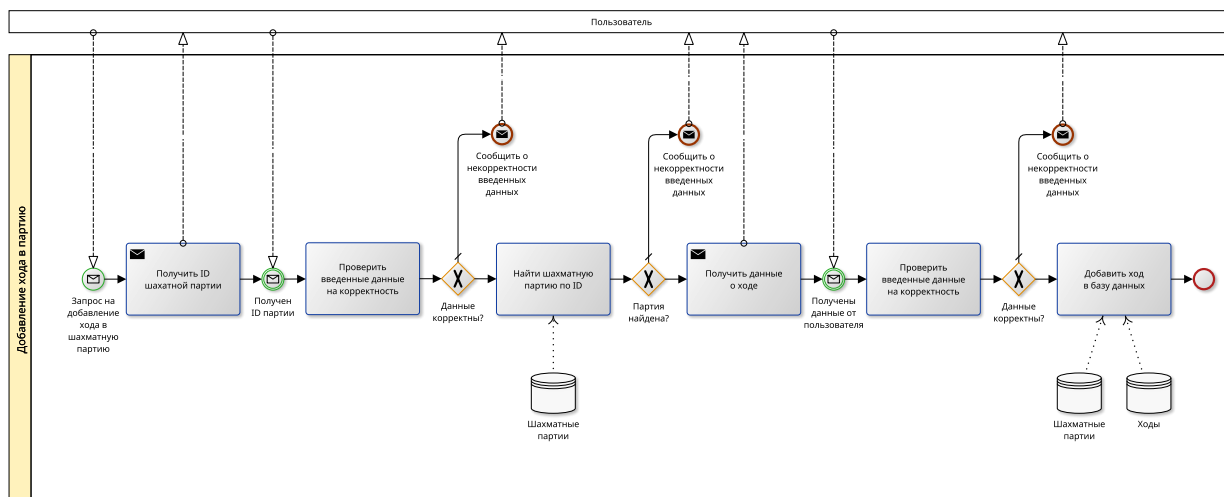


Рисунок 2.3 — Бизнес-правило добавления нового хода в партию

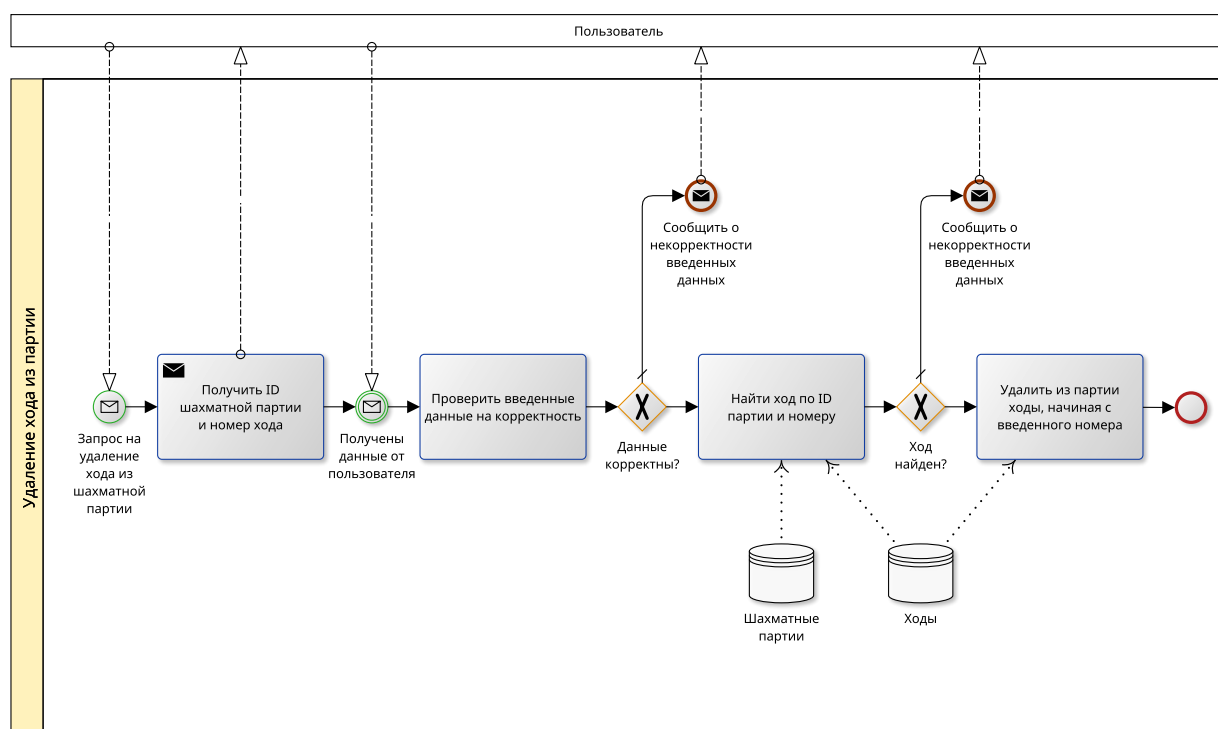


Рисунок 2.4 — Бизнес-правило удаления хода из партии

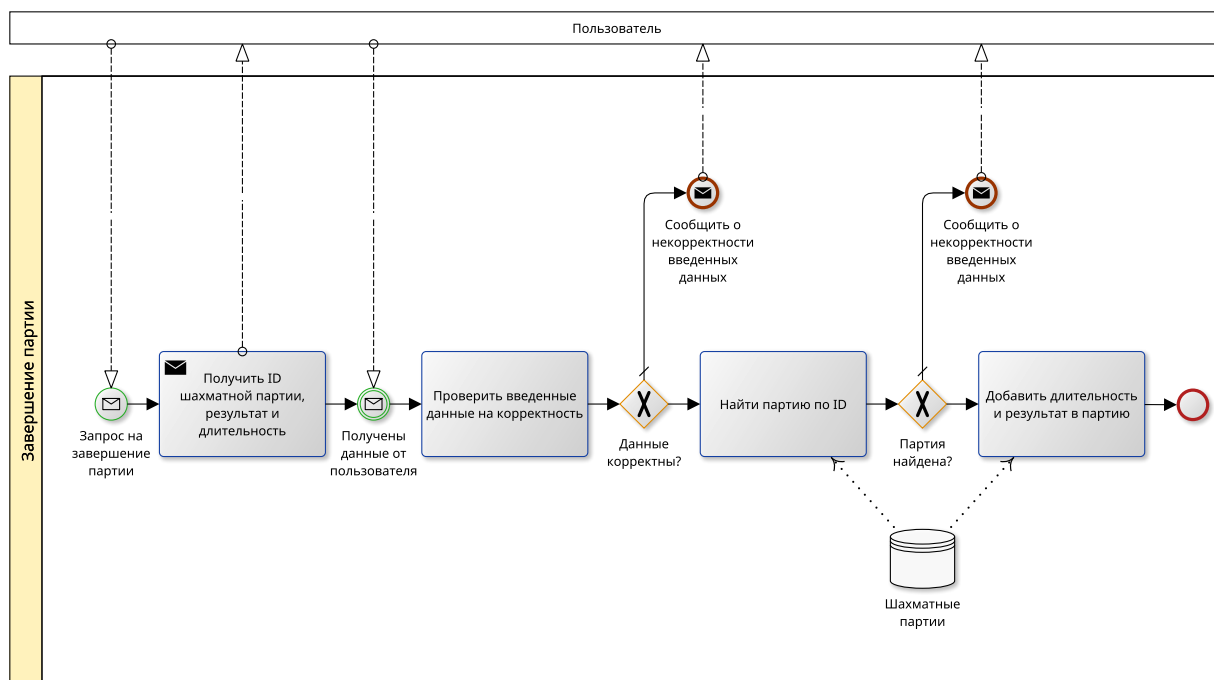


Рисунок 2.5 — Бизнес-правило завершения партии

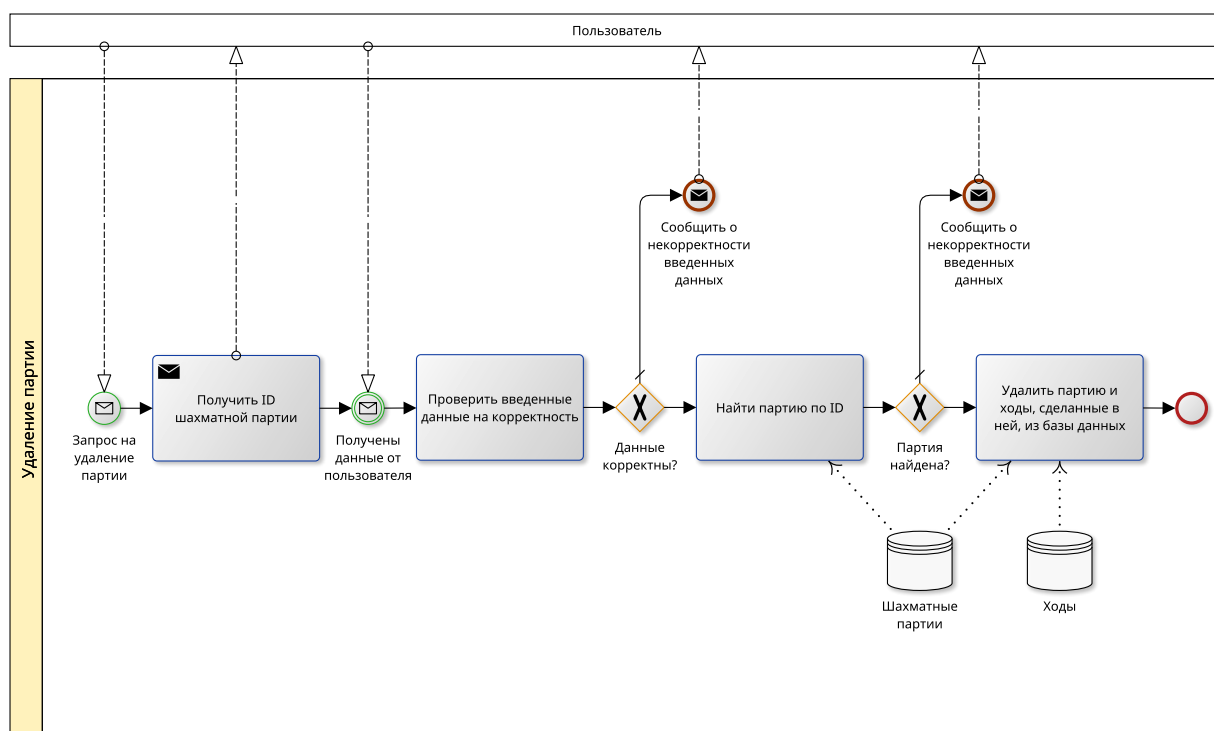


Рисунок 2.6 — Бизнес-правило удаления партии

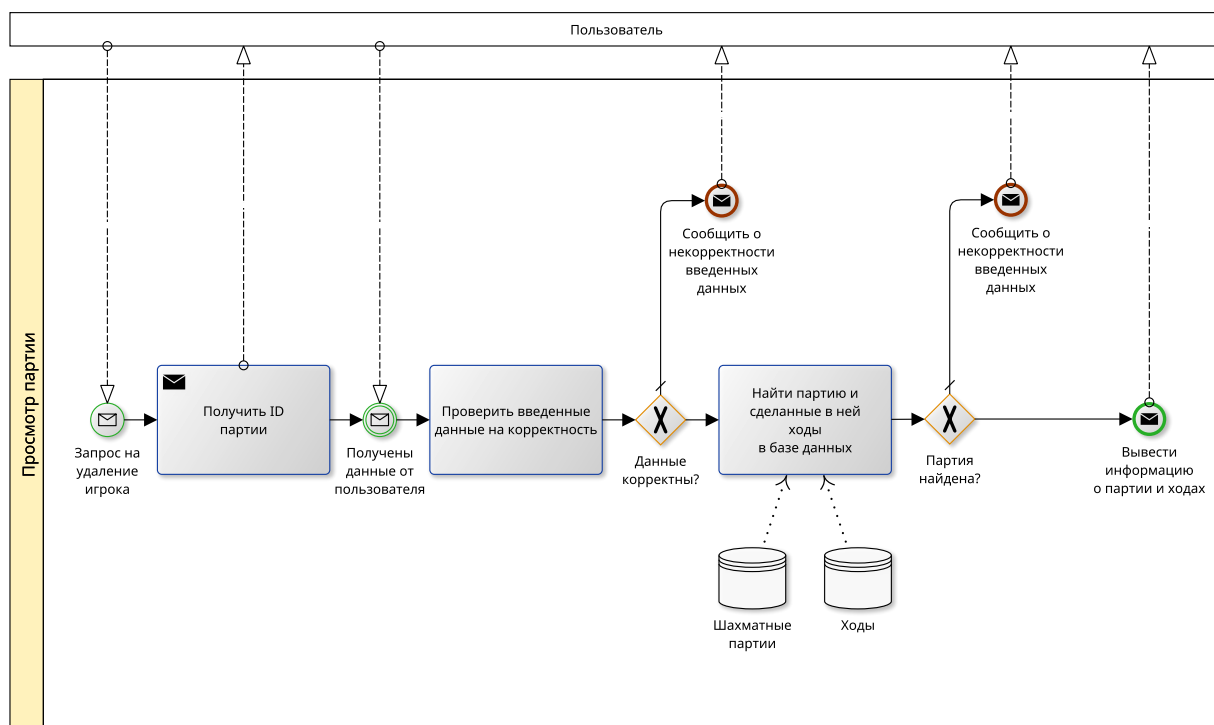


Рисунок 2.7 — Бизнес-правило просмотра партии

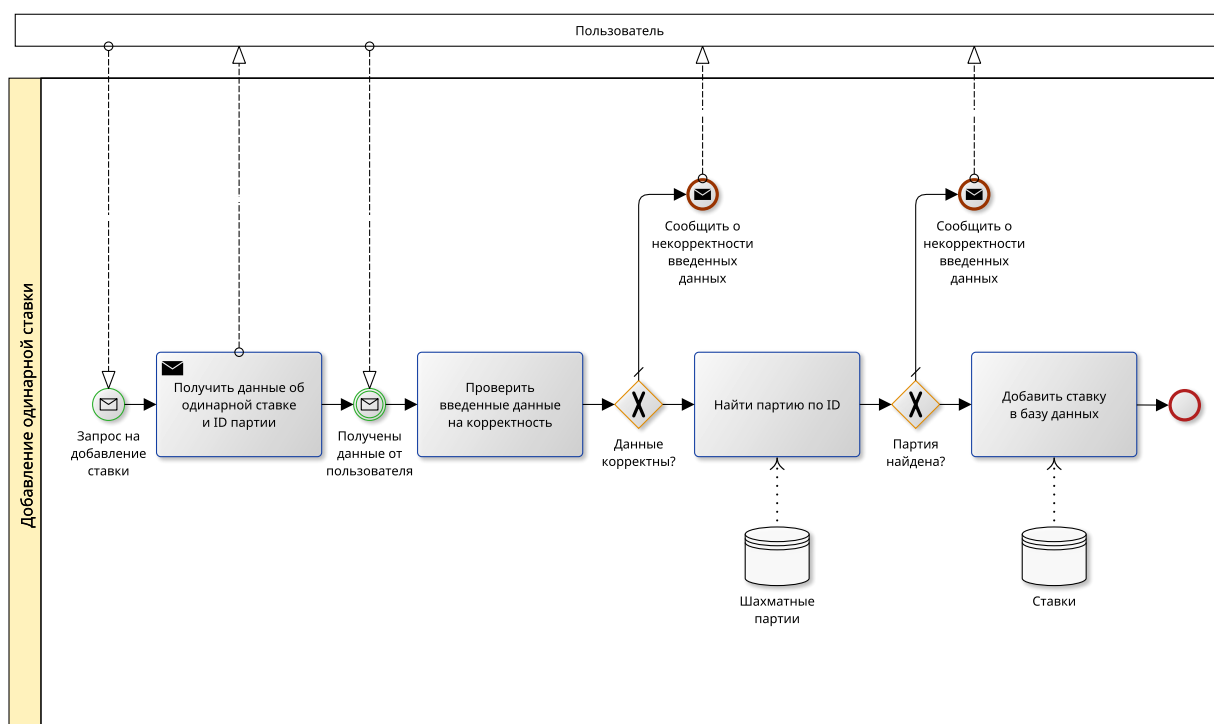


Рисунок 2.8 — Бизнес-правило добавления одинарной ставки

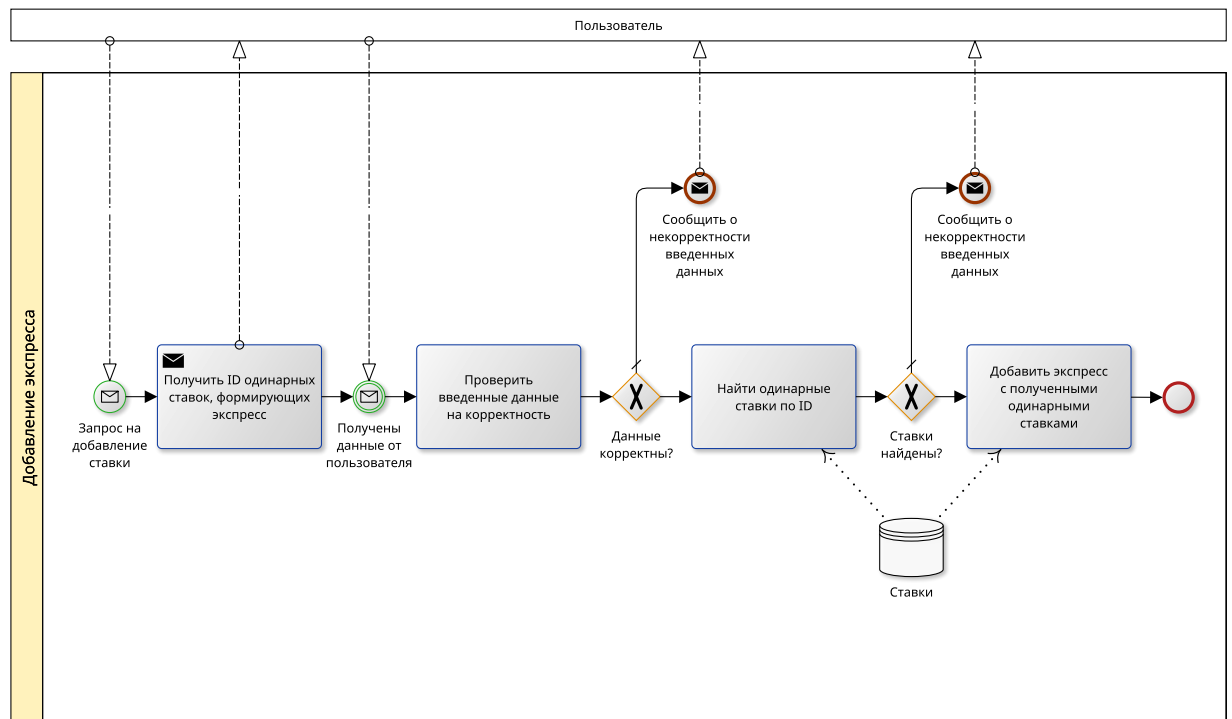


Рисунок 2.9 — Бизнес-правило добавления экспресса

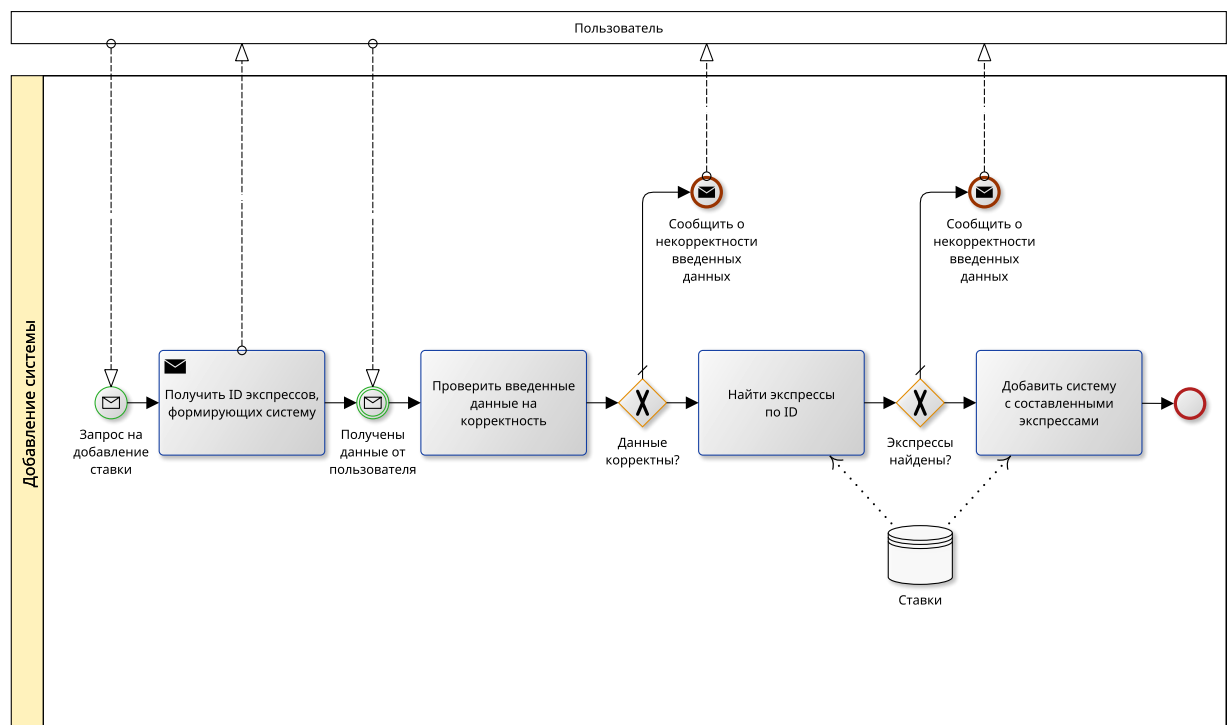


Рисунок 2.10 — Бизнес-правило добавления системы

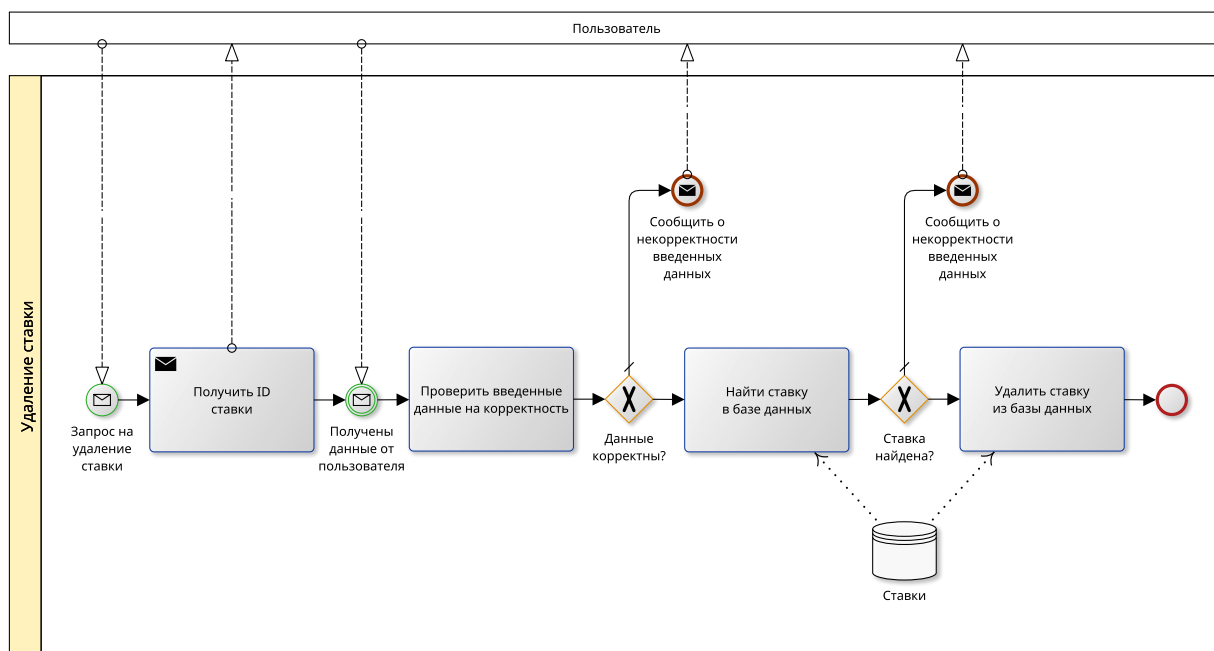


Рисунок 2.11 — Бизнес-правило удаления ставки

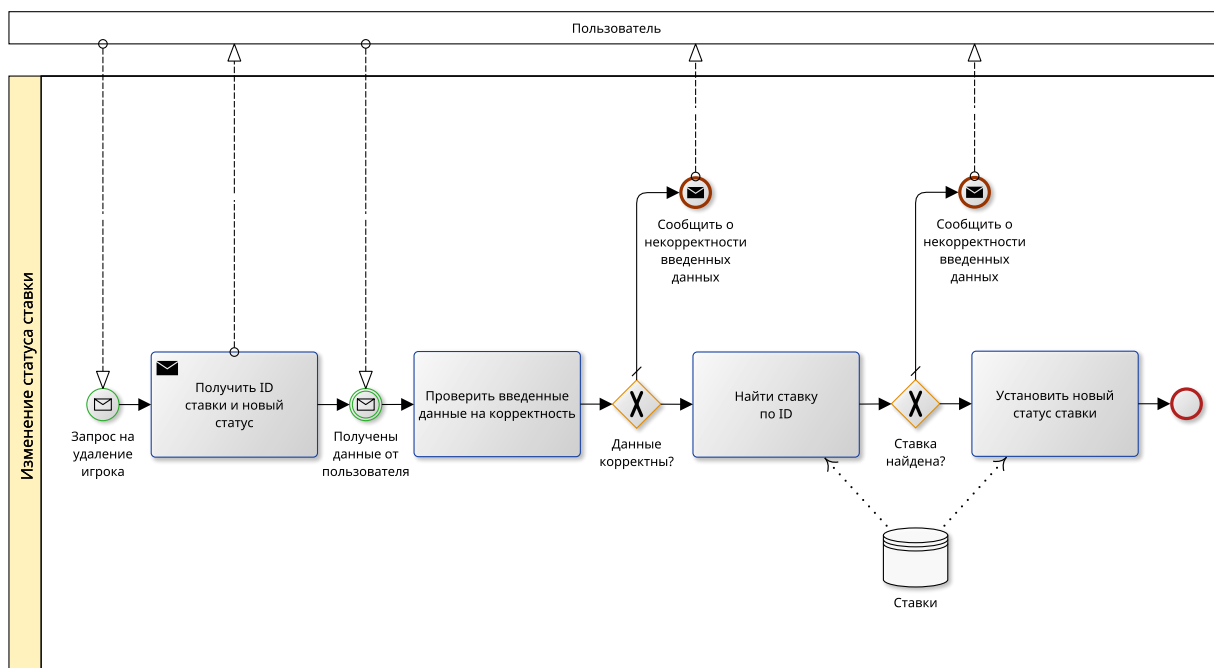


Рисунок 2.12 — Бизнес-правило изменения статуса ставки

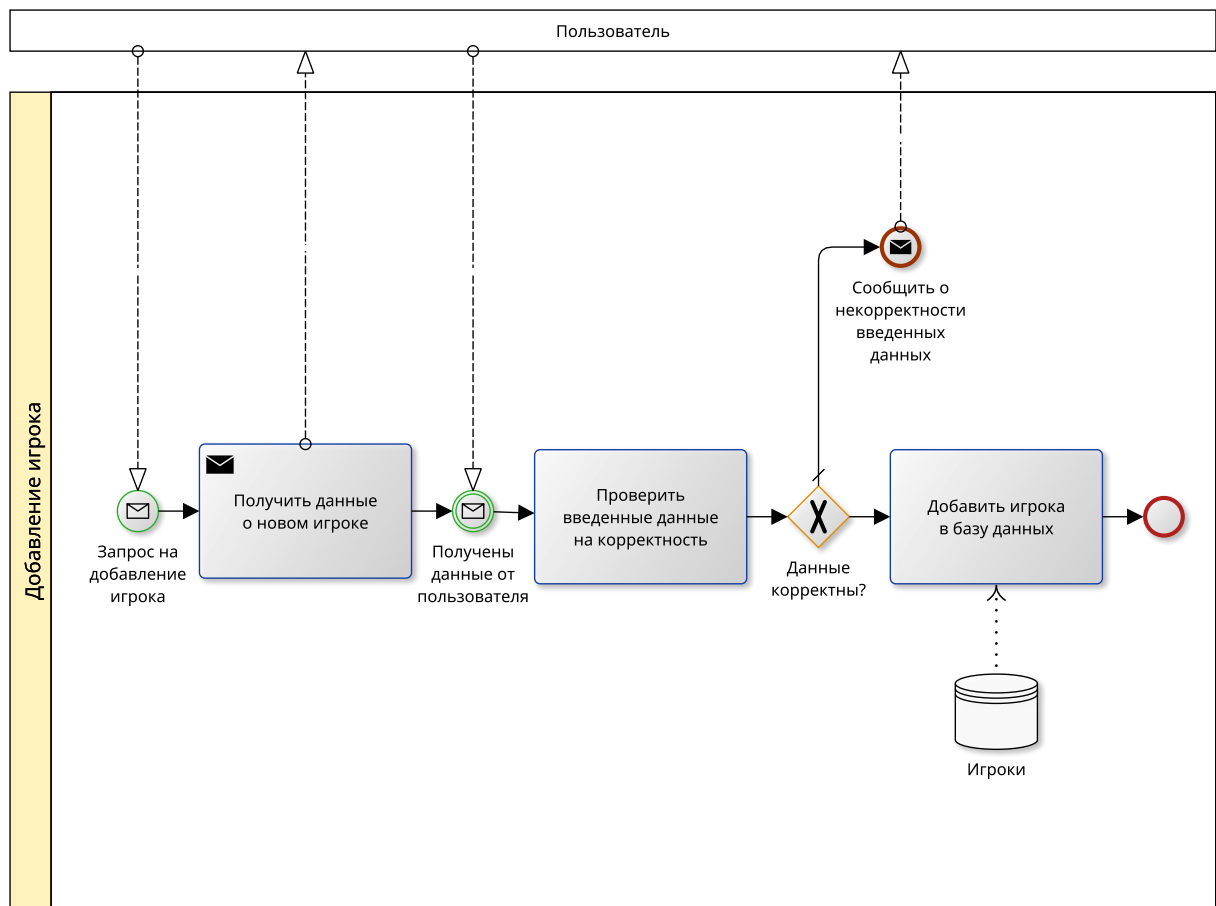


Рисунок 2.13 — Бизнес-правило добавления игрока

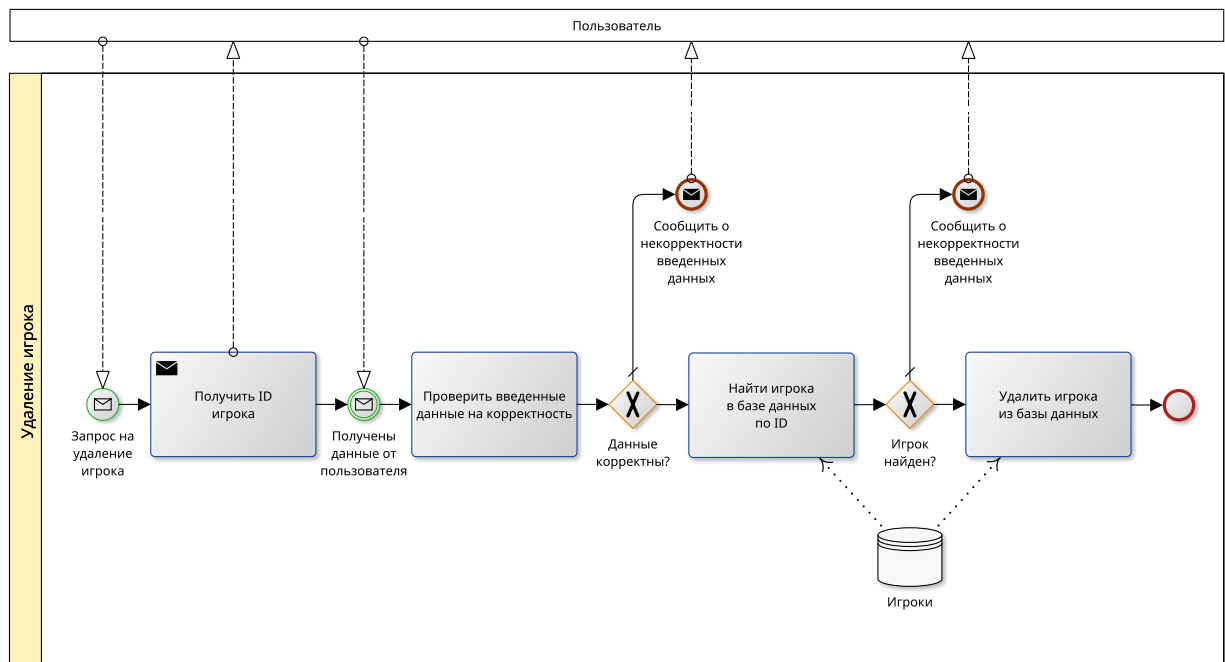


Рисунок 2.14 — Бизнес-правило удаления игрока

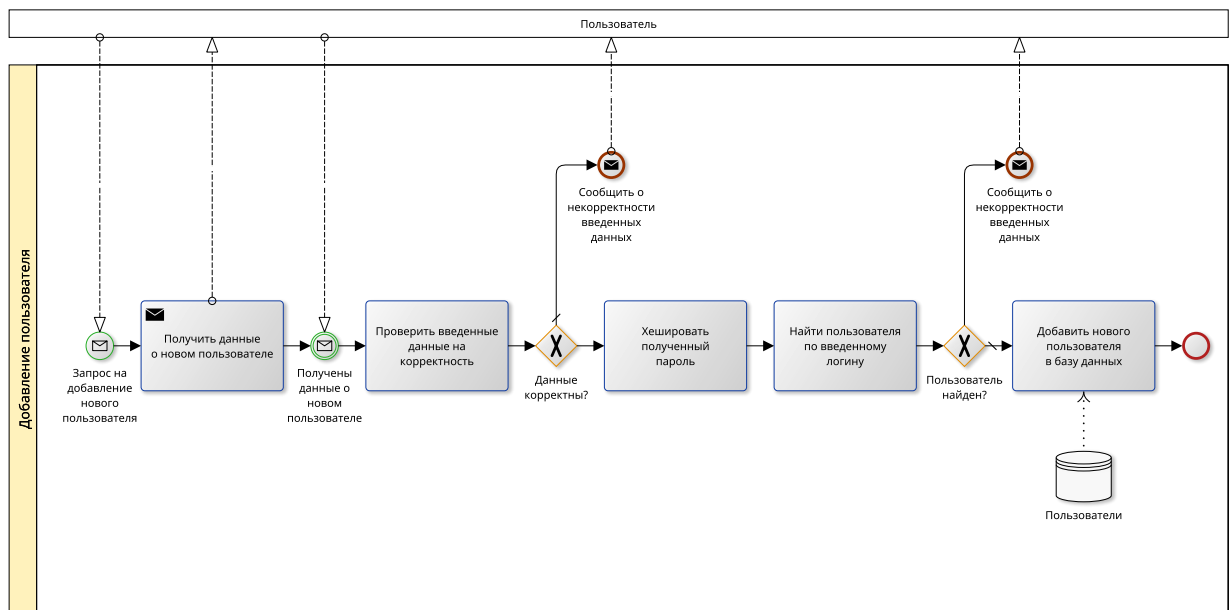


Рисунок 2.15 — Бизнес-правило добавления пользователя

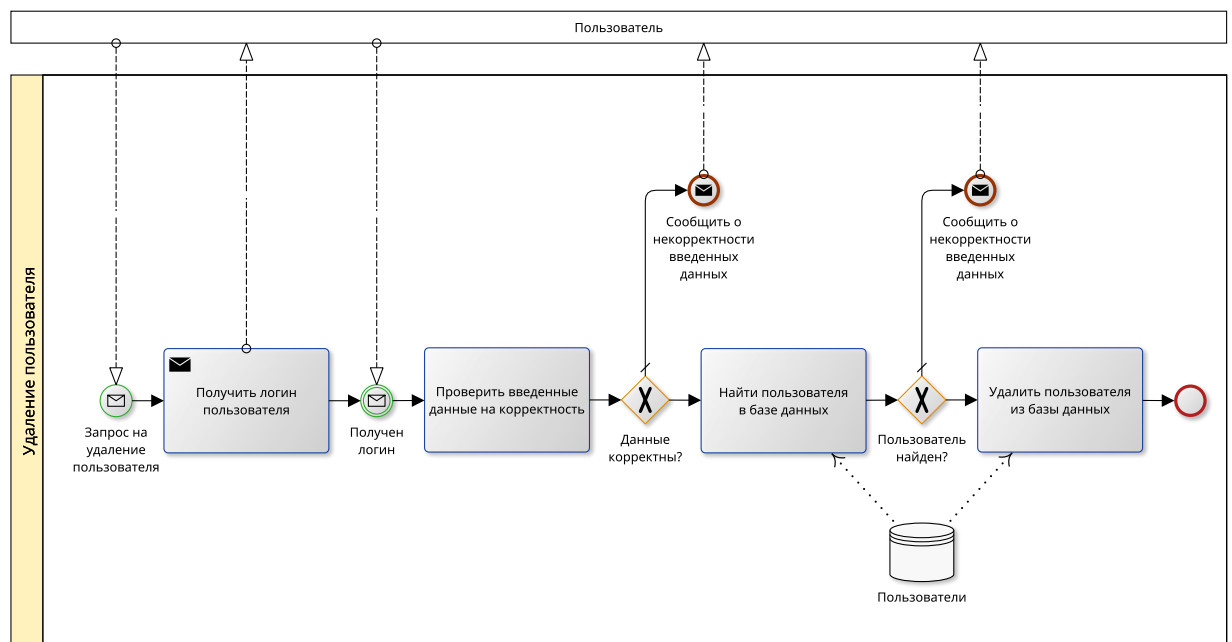


Рисунок 2.16 — Бизнес-правило удаления пользователя

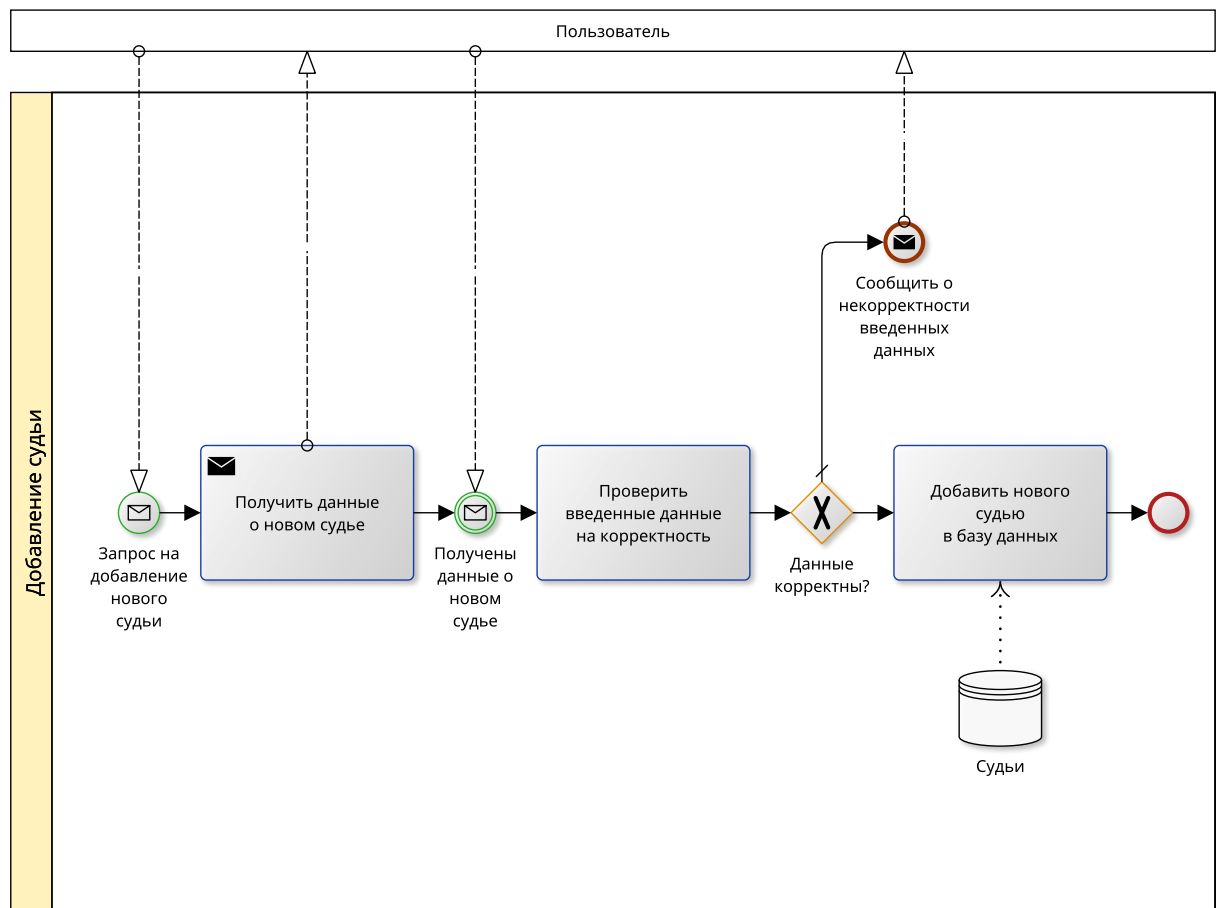


Рисунок 2.17 — Бизнес-правило добавления судьи

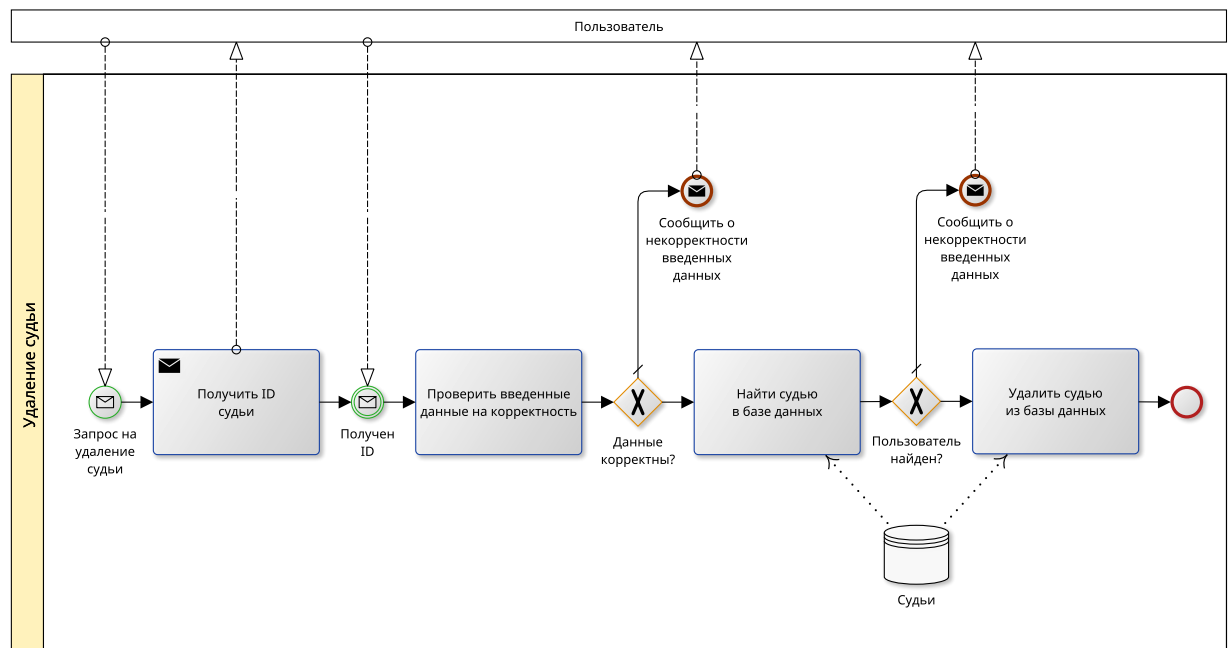


Рисунок 2.18 — Бизнес-правило удаления судьи

2.2 Проектирование базы данных

ER-диаграмма проектируемой базы данных представлена на рисунке 2.19.

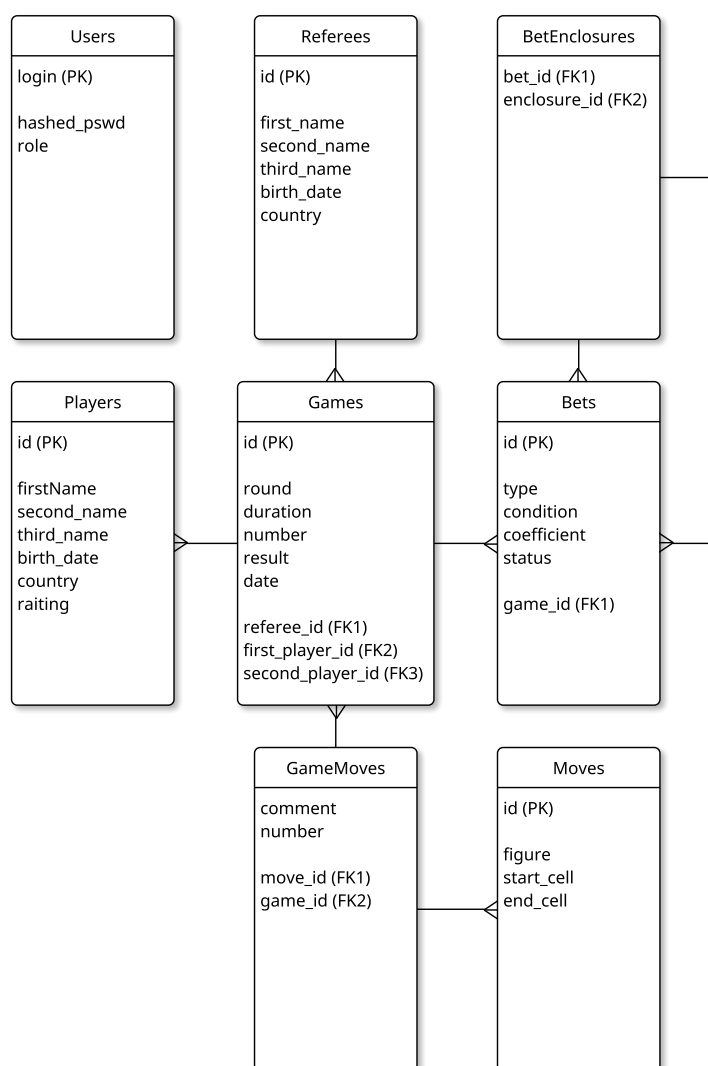


Рисунок 2.19 — ER-диаграмма проектируемой базы данных

Проектируемая база данных содержит в себе следующие таблицы:

- User — информация о пользователях системы;
- Game — информация о сыгранных шахматных партиях;
- Player — информация об игроках;
- Move — информация о ходах, сделанных в партиях;
- GameMoves — развязочная таблица для установления связи многие-ко-многим между шахматными партиями и ходами;
- Bet — информация о ставках;
- Referee — информация о судьях;
- BetEnclosures — развязочная таблица для описания вложенных ставок.

Описания полей таблиц базы данных представлены в таблицах 2.1 – 2.8.

Таблица 2.1 — Описание полей таблицы Users

Название поля	Тип	Ограничения	Значение
login	Строка	Первичный ключ Не пустая строка	Логин
hashed_pswd	Строка	Не NULL Не пустая строка	Хешированный пароль
role	Целое число	Не NULL Значения: 0 — администратор 1 — наблюдатель 2 —букмекер	Роль пользователя

Таблица 2.2 — Описание полей таблицы Referees

Название поля	Тип	Ограничения	Значение
id	Целое число	Первичный ключ	Идентификатор
first_name	Строка	Не NULL Не пустая строка	Фамилия
second_name	Строка	Не NULL Не пустая строка	Имя
third_name	Строка	—	Отчество
birth_date	Дата	Не NULL	Имя
country	Строка	Не NULL Не пустая строка	Страна

Таблица 2.3 — Описание полей таблицы BetEnclosures

Название поля	Тип	Ограничения	Значение
bet_id	Целое число	Вторичный ключ (поле id таблицы Bets) Не равен enclosure_id	Идентификатор ставки
enclosure_id	Целое число	Вторичный ключ (поле id таблицы Bets)) Не равен bet_id	Идентификатор вложенной ставки

Таблица 2.4 — Описание полей таблицы Players

Название поля	Тип	Ограничения	Значение
id	Целое число	Первичный ключ	Идентификатор
first_name	Строка	Не NULL Не пустая строка	Фамилия
second_name	Строка	Не NULL Не пустая строка	Имя
third_name	Строка	—	Отчество
birth_date	Дата	Не NULL	Имя
country	Строка	Не NULL Не пустая строка	Страна
raiting	Целое число	Не NULL Не отрицательное	Рейтинг

Таблица 2.5 — Описание полей таблицы Games

Название поля	Тип	Ограничения	Значение
id	Целое число	Первичный ключ	Идентификатор
round	Целое число	Не NULL От 1 до 8	Номер раунда
duration	Целое число	NULL или неотрицательное число	Длительность партии в секундах
number	Целое число	Не NULL Положительное	Номер партии в раунде
result	Целое число	NULL или одно из трех значений: 0 — ничья 1 — победа белых 2 — победа черных	Результат
date	Дата	—	Дата проведения
referee_id	Целое число	Вторичный ключ (поле id таблицы Referees)	Идентификатор судьи
first_player_id	Целое число	Вторичный ключ (поле id таблицы Players) Не равен second_player_id	Идентификатор первого игрока
second_player_id	Целое число	Вторичный ключ (поле id таблицы Players) Не равен first_player_id	Идентификатор второго игрока

Таблица 2.6 — Описание полей таблицы Bets

Название поля	Тип	Ограничения	Значение
id	Целое число	Первичный ключ	Идентификатор
type	Целое число	Не NULL Значения: 0 — одинарная ставка 1 — экспресс 2 — система	Тип ставки
condition	Строка	NULL или непустая строка	Условие ставки
coefficient	Вещественное число	NULL или число, большее 1	Коэффициенты ставки
status	Целое число	Не NULL Значения: 0 — не известно, 1 — ставка сработала, 2 — ставка не сработала	Статус ставки
game_id	Целое число	Вторичный ключ (поле id таблицы Games)	Идентификатор шахматной партии

Таблица 2.7 — Описание полей таблицы Moves

Название поля	Тип	Ограничения	Значение
id	Целое число	Первичный ключ	Идентификатор
figure	Целое число	Не NULL Значения: 0 — король 1 — ферзь 2 — ладья 3 — слон 4 — конь 5 — пешка	Фигура
start_cell	Строка	Не NULL Не пустая строка	Начальная клетка
end_cell	Строка	Не NULL Не пустая строка	Конечная клетка

Таблица 2.8 — Описание полей таблицы GameMoves

Название поля	Тип	Ограничения	Значение
move_id	Целое число	Вторичный ключ (поле id таблицы Moves)	Идентификатор хода
game_id	Целое число	Вторичный ключ (поле id таблицы Games)	Идентификатор шахматной партии
number	Целое число	Не NULL Положительное	Номер хода
comment	Строка	—	Комментарий

2.3 Ролевая модель проектируемой базы данных

В аналитической части были выделены следующие роли:

- неавторизованный пользователь,
- администратор,
- наблюдатель,
- букмекер.

В таблицах 2.9 – 2.12 приведена информация о доступе ролей к объектам проектируемой базы данных.

Таблица 2.9 — Доступ неавторизованного пользователя к объектам базы данных

	Таблица							
Разрешения	Users	Referees	Players	Games	Bets	Moves	Bet Enclosures	Game Moves
Select	+	-	-	-	-	-	-	-
Insert	-	-	-	-	-	-	-	-
Update	-	-	-	-	-	-	-	-
Delete	-	-	-	-	-	-	-	-

Таблица 2.10 — Доступ администратора к объектам базы данных

	Таблица							
Разрешения	Users	Referees	Players	Games	Bets	Moves	Bet Enclosures	Game Moves
Select	+	+	+	+	+	+	+	+
Insert	+	+	+	+	+	+	+	+
Update	+	+	+	+	+	+	+	+
Delete	+	+	+	+	+	+	+	+

Таблица 2.11 — Доступ наблюдателя к объектам базы данных

	Таблица							
Разрешения	Users	Referees	Players	Games	Bets	Moves	Bet Enclosures	Game Moves
Select	-	+	+	+	-	+	-	+
Insert	-	-	-	+	-	+	-	+
Update	-	-	-	+	-	+	-	+
Delete	-	-	-	+	-	+	-	+

Таблица 2.12 — Доступ букмекера к объектам базы данных

	Таблица							
Разрешения	Users	Referees	Players	Games	Bets	Moves	Bet Enclosures	Game Moves
Select	-	+	+	+	+	+	+	+
Insert	-	-	-	-	+	-	+	-
Update	-	-	-	-	+	-	+	-
Delete	-	-	-	-	+	-	+	-

2.4 Триггеры

Для корректной работы с данными необходимо реализовать следующие триггеры:

- проверка на выполнение условий ставок (главной линии, экспрессов, содержащих только главные линии; систем, содержащих экспрессы с главными линиями) после добавления завершенной партии в базу данных;
- обновление рейтинга игроков после завершения партии в базу данных;
- удаление ходов из партии.

Схема алгоритма триггера проверки на выполнение условий ставок приведена на рисунке 2.20. Схема алгоритма триггера обновления рейтинга игроков представлен на рисунках 2.21 – 2.23. Схема алгоритма удаления ходов из шахматной партии приведен на рисунках 2.24 и 2.25.

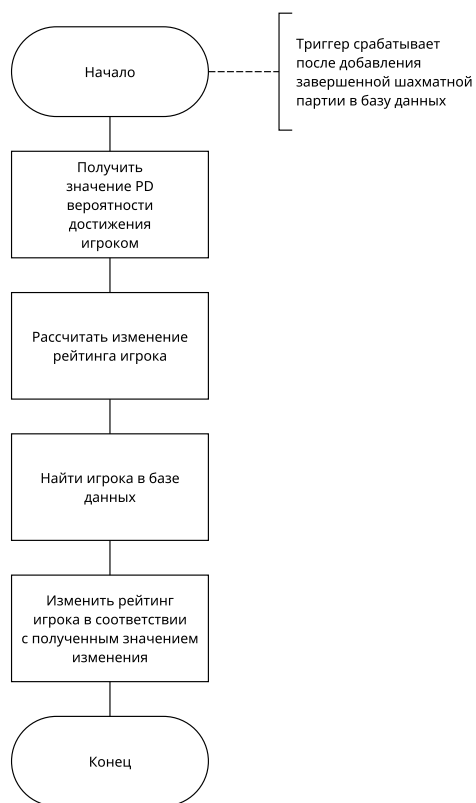


Рисунок 2.20 — Схема алгоритма триггера проверки на выполнение условий ставок

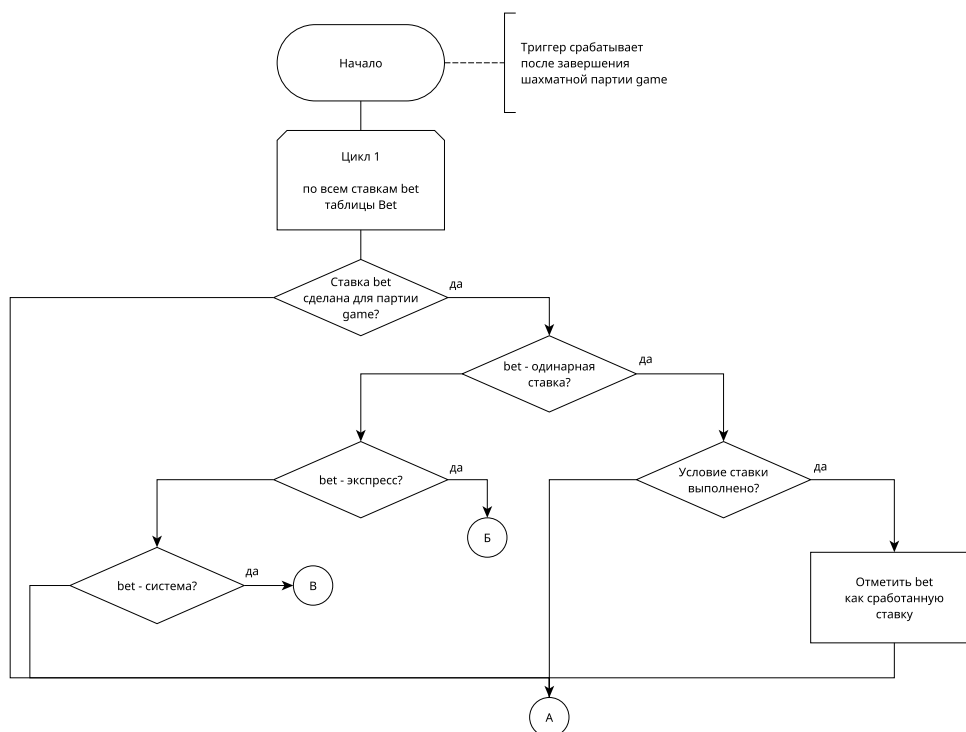


Рисунок 2.21 — Схема алгоритма триггера обновления рейтинга игроков (начало)

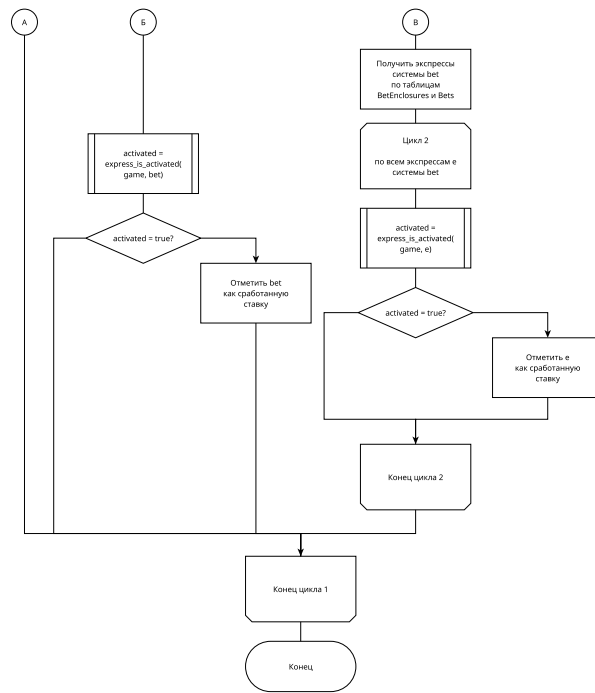


Рисунок 2.22 — Схема алгоритма триггера обновления рейтинга игроков (конец)

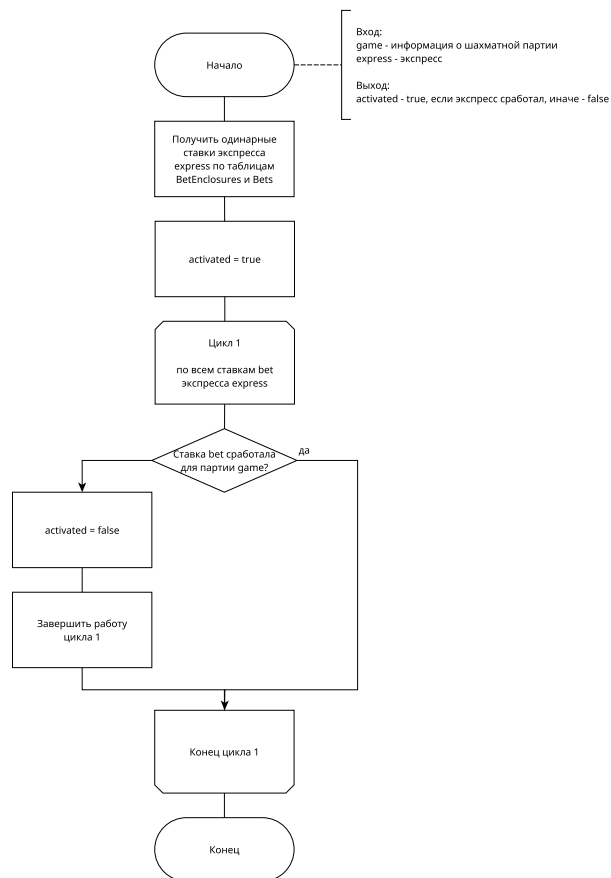


Рисунок 2.23 — Схема алгоритма функции express_is_activated

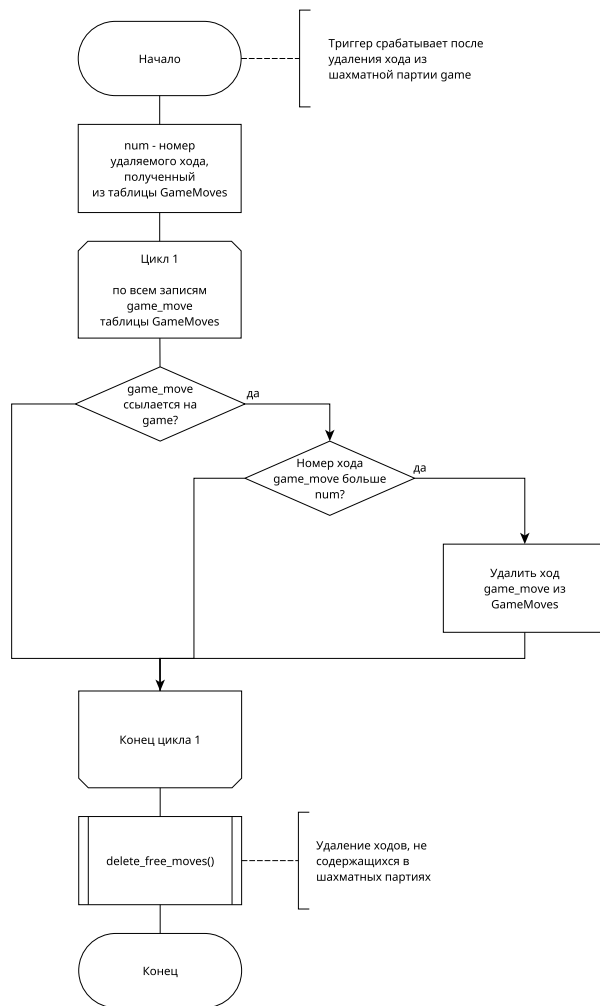


Рисунок 2.24 — Схема алгоритма удаления ходов из шахматной партии

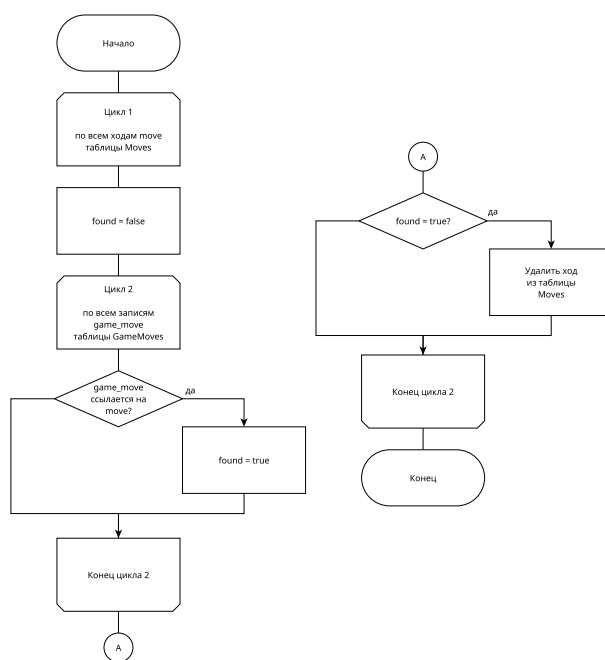


Рисунок 2.25 — Схема алгоритма функции delete_free_moves

На рисунках 2.26 – 2.28 приведены диаграммы последовательности работы триггеров.

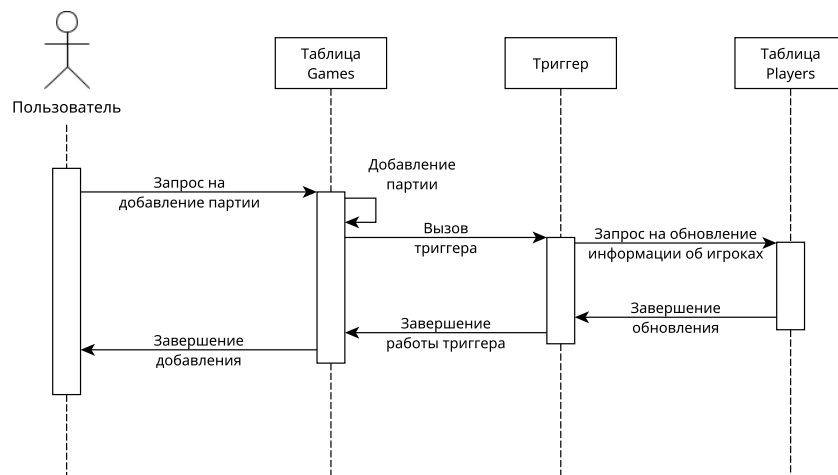


Рисунок 2.26 — Диаграмма последовательности триггера обновления рейтинга игроков

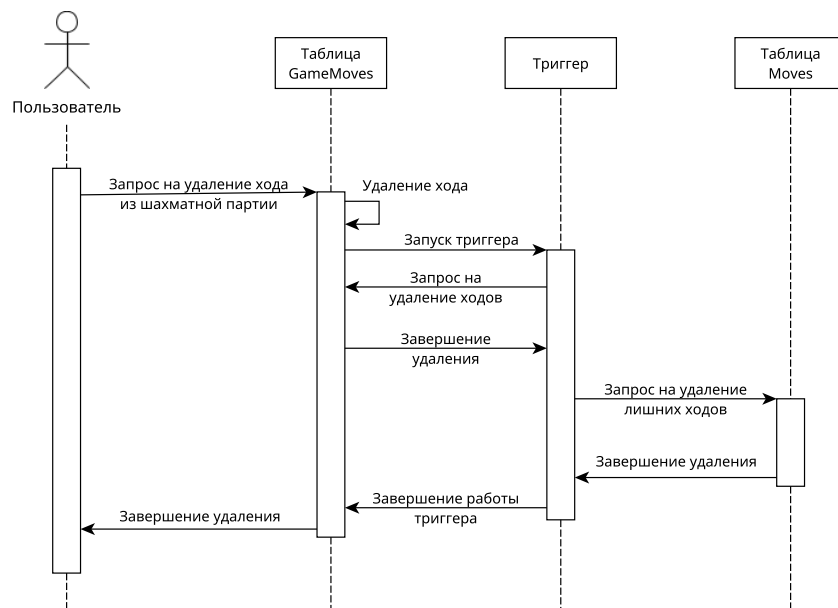


Рисунок 2.27 — Диаграмма последовательности триггера удаления ходов из шахматной партии

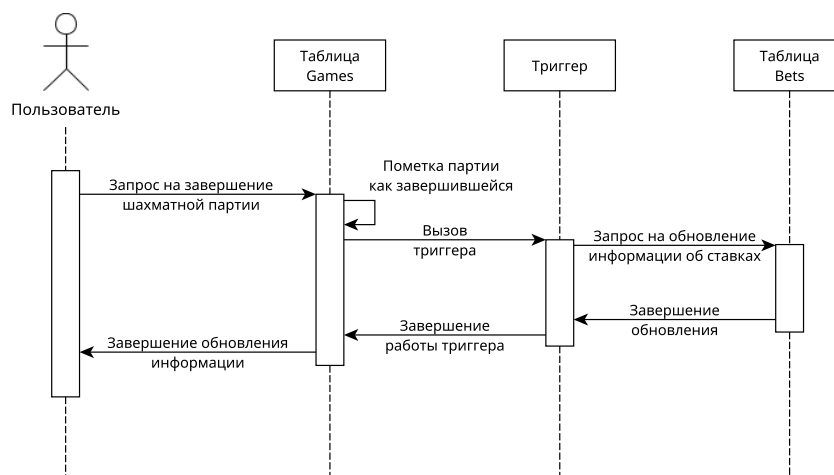


Рисунок 2.28 — Диаграмма последовательности триггера проверки на выполнение условий ставок

2.5 Структура разрабатываемого приложения

На рисунке 2.29 представлена структура разрабатываемого приложения.

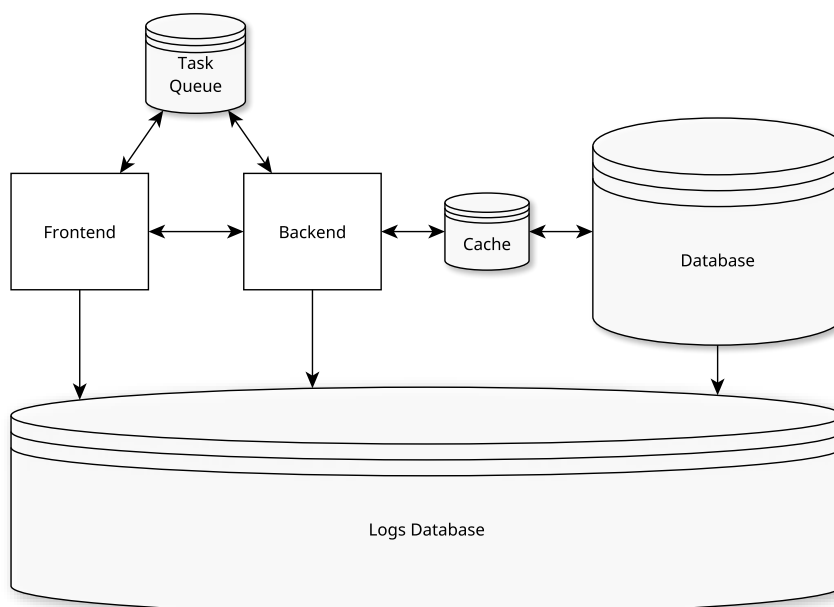


Рисунок 2.29 — Структура разрабатываемого приложения

Логирование будет осуществляться как на стороне клиента, так и на стороне сервера. В первом случае журналирование используется для мониторинга выполнения кода приложения. Логирование на стороне сервера осуществляется в рамках работы СУБД: любая манипуляция с базой данных будет отслежена и записана в лог-файл.

С точки зрения безопасности хранение журналов и базы данных на одном сервере не является лучшим решением: злоумышленник при попытке получения несанкционированного

доступа к информации может стереть или подменить логи. Для решения данной проблемы предлагается использовать второй сервер для хранения журналов.

Для уменьшения нагрузки на сервер и увеличения производительности системы предлагается использовать два кэша: между фронтендом и бекэндом и между бэкендом и базой данных. Первый кэш будет хранить очередь запросов от фронтенда, второй — результаты больших инструкций. Программа в первую очередь будет пытаться получить данные из кэша. При безрезультативном поиске система, ответственная за хранение кэша, посылает запрос серверу и после получения ответа обновляет данные.

Вывод

В конструкторском разделе были формализованы бизнес-правила приложения и спроектирована база данных. Были описаны ролевая модель и ограничения базы данных. Были разработаны схемы алгоритмов триггеров, необходимых для корректной работы системы. Была описана структура разрабатываемого приложения.

3 Технологическая часть

3.1 Выбор средств реализации

Для реализации приложения был выбран высокоуровневый язык Java, поскольку он поддерживает объектно-ориентированную парадигму программирования и позволяет писать приложения, запускаемые в любой системе с помощью виртуальной Java-машины [12].

Для хранения информации, связанной с предметной областью, будет использована клиент-серверная СУБД. В таблице 3.1 представлен сравнительный анализ наиболее популярных клиент-серверных СУБД.

Таблица 3.1 — Сравнительный анализ клиент-серверных СУБД

Критерий	СУБД		
	PostgreSQL [13]	MongoDB [14]	Oracle [15]
Тип	Реляционная	Документоориентированная	Реляционная
Открытый исходный код	+	+	-
Операционная система сервера	Windows Unix Linux OS X	Windows Unix Linux	Windows Linux OS X
Поддержка SQL	+	-	+
Поддержка триггеров	+	-	+

Для работы с базой данных была выбрана система PostgreSQL, поскольку она имеет открытый исходный код, доступна под наиболее популярные операционные системы и имеет поддержку триггеров.

Приложение должно обеспечивать хранение логов в отдельной базе данных. В качестве СУБД будет использована СУБД временных рядов, поскольку, в отличие от других систем, она оптимизирована для быстрого приема запросов: скорость загрузки данных не уменьшается со временем и остается стабильной [16]. В таблице [?] представлен сравнительный анализ наиболее популярных СУБД временных рядов.

Таблица 3.2 — Сравнительный анализ СУБД временных рядов

Критерий	СУБД		
	InfluxDB [17]	Prometheus [18]	Graphite [19]
Тип по способу доступа	Клиент-серверная	Клиент-серверная	Клиент-серверная
Открытый исходный код	+	+	+
Операционная система сервера	Linux OS X	Windows Linux	Linux Unix
Поддерживаемые типы данных	Числа и строки	Числа	Числа

Для логирования была выбрана СУБД InfluxDB, поскольку она является клиент-серверной и поддерживает такие типы данных, как числа и строки. Для логирования СУБД PostgreSQL в InfluxDB был выбран плагин Telegraf [20].

В качестве СУБД для кэширования информации и организации очереди запросов предлагается использовать встраиваемые системы «ключ-значение». В таблице 3.3 представлен сравнительный анализ наиболее популярных встраиваемых СУБД «ключ-значение».

Таблица 3.3 — Сравнительный анализ встраиваемых СУБД «ключ-значение»

Критерий	СУБД	
	Memcached [21]	Redis [22]
Открытый исходный код	+	+
Операционная система	Windows Unix Linux OS X	Windows Linux OS X
Поддержка скриптов	-	+
Поддержка транзакций	-	+

Для кэширования и организации очереди запросов была выбрана СУБД Redis, поскольку она поддерживает пользовательские скрипты и транзакции.

3.2 Описание сущностей и ограничений целостности базы данных

В листингах 3.1–3.8 представлены описания сущностей и ограничений целостности базы данных.

Листинг 3.1 — Описание сущности users

```
1 create table if not exists users
2 (
3     login text primary key,
4     hashed_pswd text not null check (hashed_pswd != ''),
5     role int not null check (role <= 2 and role >= 0)
6 );
```

Листинг 3.2 — Описание сущности referees

```
1 create table if not exists referees
2 (
3     id int primary key,
4     first_name text not null check (first_name != ''),
5     second_name text not null check (second_name != ''),
6     third_name text,
7     birth_date date not null,
8     country text not null check (country != '')
9 );
```

Листинг 3.3 — Описание сущности players

```
1 create table if not exists players
2 (
3     id int primary key,
4     first_name text not null check (first_name != ''),
5     second_name text not null check (second_name != ''),
6     third_name text,
7     birth_date date not null,
8     country text not null check (country != ''),
9     raiting int not null check (raiting >= 0)
10 );
```

Листинг 3.4 — Описание сущности moves

```
1 create table if not exists moves
2 (
3     id int primary key,
4     figure int not null check (figure >= 0 and figure <= 5),
5     start_cell text not null check (start_cell != ''),
6     end_cell text not null check (end_cell != '')
7 );
```

Листинг 3.5 — Описание сущности games

```
1 create table if not exists games
2 (
3     id int primary key,
4     round int not null check (round >= 1 and round <= 8),
5     duration int,
6     number int not null check (number > 0),
7     result int check (result >= 0 and result <= 2),
8     date date,
9     referee_id int references referees (id) on delete cascade,
10    first_player_id int references players (id) on delete cascade,
11    second_player_id int references players (id) on delete cascade
12 );
13
14 alter table games add constraint first_player_id check (
15     first_player_id != second_player_id);
16 alter table games add constraint second_player_id check (
17     first_player_id != second_player_id);
```

Листинг 3.6 — Описание сущности bets

```
1 create table if not exists bets
2 (
3     id int primary key,
4     type int not null check (type >= 0 and type <= 2),
5     condition text check (condition = ''),
6     coefficient real check (coefficient > 1),
7     status int not null check (status >= 0 and status <= 2),
8     game_id int references games (id) on delete cascade
9 );
```

Листинг 3.7 — Описание сущности game_moves

```

1 create table if not exists game_moves
2 (
3     game_id int references games (id) on delete cascade,
4     move_id int references moves (id) on delete cascade,
5     number int not null check (number > 0),
6     comment text
7 );

```

Листинг 3.8 — Описание сущности bet_enclosures

```

1 create table if not exists bet_enclosures
2 (
3     bet_id int references bets (id) on delete cascade,
4     enclosure_id int references bets (id) on delete cascade
5 );
6
7 alter table bet_enclosures add constraint bet_id check (bet_id !=
8     enclosure_id);
9
10 alter table bet_enclosures add constraint enclosure_id check (bet_id
11     != enclosure_id);

```

3.3 Реализация триггеров

В листингах 3.9–3.12 представлены реализованные триггеры и функции, необходимые для их выполнения.

Листинг 3.9 — Триггер удаления ходов из шахматной партии

```

1 create or replace function remove_next_moves() returns trigger as
2     $remove_next_moves_trigger$
3     begin
4         delete from game_moves where move_id >= old.move_id and game_id
5             == old.game_id;
6         delete from moves where id not in (select move_id from
7             game_moves);
8     end;
9     $remove_next_moves_trigger$
10 language plpgsql;
11
12 create or replace trigger remove_next_moves_trigger
13 after delete on game_moves
14 execute function remove_next_moves();

```

Листинг 3.10 — Триггер проверки выполнения условий ставок

```

1 create or replace function update_bets_status() returns trigger as
2   $update_bets_status_trigger$
3   declare
4     bet record;
5     express_status int;
6     express record;
7   begin
8     for bet in select * from bets
9       loop
10        if bet.type == 0 then
11          if bet.game_id == old.id then
12            if is_main_line(bet.condition) then
13              if is_achieved(bet.condition, old.result) then
14                update bets set status = 1 where id = bet.id;
15              else
16                update bets set status = 2 where id = bet.id;
17              end if;
18            end if;
19          end if;
20        elsif bet.type == 1 then
21          express_status = new_express_status(bet, game);
22          if express_status != bet.status then
23            update bets set status = express_status where id =
24              bet.id;
25          end if;
26        elsif bet.type == 2 then
27          for express in
28            select enclosure_id as id, type, condition,
29              coefficient, status, game_id
30            from game_moves gm join bets b on enclosure_id = id
31            where bet_id = bet.id
32          loop
33            express_status = new_express_status(express, old);
34            if express_status != express.status then
35              update bets set status = express_status where id =
36                express.id;
37            end if;
38          end loop;
39        end if;
40      end loop;
41      return null;
42    end;
43  $update_bets_status_trigger$
44 language plpgsql;
45
46 create trigger update_bets_status_trigger
47 after update of result on games
48 for each row
49 execute function update_bets_status();

```

Листинг 3.11 — Триггер обновления рейтинга игроков (начало)

```

1 create or replace function update_ratings() returns trigger as
2   $update_ratings_trigger$
3     declare
4       prob1 real;
5       prob2 real;
6       raiting1 int;
7       raiting2 int;
8       new_raiting1 int;
9       new_raiting2 int;
10      delta1 int;
11      delta2 int;
12      raiting_delta1 int;
13      raiting_delta2 int;
14    begin
15      select raiting into raiting1 from players where id = new.
16        first_player_id;
17      select raiting into raiting2 from players where id = new.
18        second_player_id;
19      if (raiting1 > raiting2) then
20        select weak_player_win_prob(raiting1 - raiting2) into prob2;
21        prob1 = 1.0 - prob2;
22      else
23        select weak_player_win_prob(raiting2 - raiting1) into prob1;
24        prob2 = 1.0 - prob1;
25      end if;
26      if new.result == 0 then
27        delta1 = 0.5 - prob1;
28        delta2 = 0.5 - prob2;
29      elsif new.result == 1 then
30        delta1 = 1 - prob1;
31        delta2 = -1 * prob2;
32      else
33        delta1 = -1 * prob1;
34        delta2 = 1 - prob2;
35      end if;
36      if raiting1 <= 2400 then
37        raiting_delta1 = delta1 * 20;
38      else
39        raiting_delta1 = delta1 * 10;
40      end if;
41      if raiting2 <= 2400 then
42        raiting_delta2 = delta2 * 20;
43      else
44        raiting_delta2 = delta2 * 10;
45      end if;
46      if raiting_delta1 > 700 then
47        raiting_delta1 = 700;
48      end if;

```

Листинг 3.12 — Триггер обновления рейтинга игроков (конец)

```

1      if raiting_delta2 > 700 then
2          raiting_delta2 = 700;
3      end if;
4      update players set raiting = raiting + raiting_delta1 where id
          = new.first_player_id;
5      update players set raiting = raiting + raiting_delta2 where id
          = new.second_player_id;
6      return new;
7  end;
8  $update_raittings_trigger$
9  language plpgsql;
10
11 create trigger update_raittings_trigger
12 after update of result on games
13 for each row
14 execute function update_raittings();

```

3.4 Описание ролевой модели на уровне базы данных

В листингах 3.13–3.16 представлено описание ролевой модели на уровне базы данных.

Листинг 3.13 — Описание роли неавторизованного пользователя

```

1 create user unauthorized with encrypted password 'unauthorized';
2 grant select on users to unauthorized;

```

Листинг 3.14 — Описание роли администратора

```

1 create user administrator with encrypted password 'administrator';
2 grant select, update, delete, insert on bet_enclosures to
    administrator;
3 grant select, update, delete, insert on bets to administrator;
4 grant select, update, delete, insert on bet_enclosures to
    administrator;
5 grant select, update, delete, insert on game_moves to administrator;
6 grant select, update, delete, insert on games to administrator;
7 grant select, update, delete, insert on moves to administrator;
8 grant select, update, delete, insert on players to administrator;
9 grant select, update, delete, insert on referees to administrator;

```

Листинг 3.15 — Описание роли наблюдателя

```
1 create user spectator with encrypted password 'spectator';  
2 grant select on referees, players to spectator;  
3 grant select, update, delete, insert on games to spectator;  
4 grant select, update, delete, insert on moves to spectator;  
5 grant select, update, delete, insert on game_moves to spectator;
```

Листинг 3.16 — Описание роли букмекера

```
1 create user bookmaker with encrypted password 'bookmaker';  
2 grant select on referees, players, games, moves, game_moves to  
   bookmaker;  
3 grant select, update, delete, insert on bets to bookmaker;  
4 grant select, update, delete, insert on bet_enclosures to bookmaker;
```

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Regulations for the FIDE World Cup 2023 [Электронный ресурс] — Режим доступа: <https://handbook.fide.com/files/handbook/WorldCup2023Regulations.pdf> (дата обращения 22.08.23)
2. FIDE Rating Regulations effective from 1 January 2022 [Электронный ресурс] — Режим доступа: <https://handbook.fide.com/chapter/B022022> (дата обращения 23.08.23)
3. Elo, Arpad E. The Proposed USCF Rating System, Its Development, Theory, and Applications [Электронный ресурс] — Режим доступа: http://uscf1-nyc1.aodhosting.com/CL-AND-CR-ALL/CL-ALL/1967/1967_08.pdf#page=26 (дата обращения 23.08.23)
4. Как делать ставки на спорт в букмекерских конторах [Электронный ресурс] — Режим доступа: <https://journal.tinkoff.ru/you-bet/> (дата обращения 23.08.23)
5. Как ставить на шахматы: особенности игры, советы игрокам [Электронный ресурс] — Режим доступа: <https://bookmaker-ratings.ru/wiki/kak-stavit-na-shahmaty-osobennosti-igry-sovety-igrokam/> (дата обращения 23.08.23)
6. Scid — Chess Database Software [Электронный ресурс] — Режим доступа: <https://scid.sourceforge.net/> (дата обращения 03.04.23)
7. Chessgames.com: Chess Games Database & Community [Электронный ресурс] — Режим доступа: <https://www.chessgames.com/> (дата обращения 03.04.23)
8. Chess Games Database Online — 365Chess.com [Электронный ресурс] — Режим доступа: <https://www.365chess.com/> (дата обращения 03.04.23)
9. ICS Games Database [Электронный ресурс] — Режим доступа: <https://www.ficsgames.org/> (дата обращения 03.04.23)
10. Дейт К.Дж. Введение в системы баз данных, 8-е изд.: Пер. с англ. — СПб.: ООО «Диалектика», 2019. — 1328 с.: ил. — Парал. тит. англ.
11. Кузнецов С.Д. Базы данных [Электронный ресурс] — Режим доступа: [https://k0d.cc/storage/books/Databases/Базы данных \(Кузнецов 2020\).pdf](https://k0d.cc/storage/books/Databases/Базы данных (Кузнецов 2020).pdf) (дата обращения 24.08.23)

12. Java [Электронный ресурс] — URL: <https://www.java.com/en/> (дата обращения 19.09.23)
13. PostgreSQL: The World's Most Advanced Open Source Relational Database [Электронный ресурс] — URL: <https://www.postgresql.org/> (дата обращения 19.09.23)
14. Microsoft Data Platform [Электронный ресурс] — URL: <https://www.microsoft.com/en-us/sql-server> (дата обращения 19.09.23)
15. Oracle | Cloud Applications and Cloud Platform [Электронный ресурс] — URL: <https://www.oracle.com/> (дата обращения 19.09.23)
16. Mueen A., Keogh E., Zhu Q., Cash S., Westover B. Exact Discovery of Time Series Motifs [Электронный ресурс] — URL: <https://web.archive.org/web/20100625200233/https://www.cs.ucr.edu/~eamonn/EM.pdf> (дата обращения 19.09.23)
17. InfluxDB Times Series Data Platform | InfluxData [Электронный ресурс] — URL: <https://www.influxdata.com/> (дата обращения 19.09.23)
18. Prometheus — Monitoring system & time series database [Электронный ресурс] — URL: <https://prometheus.io/> (дата обращения 19.09.23)
19. Graphite is a scalable monitoring system for timeseries data [Электронный ресурс] — URL: <https://grafana.com/oss/graphite/> (дата обращения 19.09.23)
20. Telegraf: The plugin-driven server agent for collecting & reporting metrics [Электронный ресурс] — URL: <https://github.com/influxdata/telegraf> (дата обращения 19.09.23)
21. memcached — a distributed memory object caching system [Электронный ресурс] — URL: <https://memcached.org/> (дата обращения 19.09.23)
22. Redis [Электронный ресурс] — URL: <https://redis.io/> (дата обращения 19.09.23)