

Федеральное государственное автономное
образовательное учреждение высшего образования

Университет ИТМО

**Курсовая работа по дисциплине
"Информационные системы и базы данных".**

Этап 1

Выполнили:

Марков Петр Денисович
Кривоносов Егор Дмитриевич

Группа: Р33111

Преподаватель:

Николаев Владимир Вячеславович

2021 г.

Persona 5. Идеальный мир доктора Маруки.



Описание предметной области

В Persona 5 существует один персонаж — психолог Маруки. У него есть уникальная способность — он способен исполнять мечты людей. Он может исполнять как свои, так и мечты других людей. Будучи человеком с добрыми намерениями он способен создать новый мир, исполнив желания всех людей, а также менять восприятие конкретного человека, его мировоззрение.

В самой игре половину работы за доктора делает его Персона. Именно поэтому в мире существуют определенные недочеты, которые доктор с помощью своих сил пытается сделать незаметными для окружающих людей (люди попросту не замечают странных вещей). Благодаря этой уязвимости Джокер и смог восстановить память своей команде. Поэтому мы считаем, что такой процесс нужно оптимизировать и доработать, чтобы не было никаких изъянов.

[Ссылка №1](#) [Ссылка №2](#) [Ссылка №3](#)

Бизнес-процессы

Наша система

Качества человека (тест Лири)

- Умеет настоять на своем
- Обладает чувством достоинства
- Независимый
- Строгий, но справедливый
- Искренний
- Способен признать свою неправоту
- Способный к сотрудничеству, взаимопомощи
- Стремится ужиться с другими
- Дружелюбный, доброжелательный
- Внимательный, ласковый
- Деликатный
- Отзывчивый
- Бескорыстный
- Способен вызывать восхищение
- Пользуется у других уважением
- Обладает талантом руководителя
- Дружелюбный

- Уверен в себе
- Самоуверен, напорист
- Деловитый, практичный
- Стойкий и упорный, где надо
- Беспристрастный
- Открытый, прямолинейный
- Скромный
- Дорожит мнением окружающих
- Общительный
- Добросердечный
- Добрый, вселяющий уверенность
- Нежный, мягкосердечный
- Бескорыстный, щедрый
- Откровенный
- Инициативный
- Великодушный



Стержневые

- Мир
 - ид
 - реальный_мир_ид
 - описание
- Мир Персоны
 - ид
 - реальный_мир_ид
 - метаверс_ид (искаженный мир)
- Локации
 - ид
 - реальный_мир_ид
 - реальная_локация_ид
 - название
 - описание
- Достопримечательность
 - ид
 - реальная_достопримечательность_ид
 - локация_ид
 - название
 - описание
 - дата_постройки
- События
 - ид
 - локация_ид
 - название
 - описание
 - дата_события
 - хорошее_или_плохое ? (true/false)

- Качества
 - ид
 - название
 - хорошее_или_плохое ? (true/false)
- Эмоции
 - ид
 - название
 - описание
 - хорошее_или_плохое ? (true/false)
- Люди
 - ид
 - имя
 - фамилия
 - реальный_человек ? (true/false)
 - пол
 - раса
 - ориентация
 - темперамент
 - социальный статус
 - дата рождения (может не быть)
 - дата смерти (может не быть)
- Желание
 - ид
 - тень_ид
 - человек_желающий_это_ид
- Тень
 - ид
 - зона_ид
 - реальный_человек_ид
 - прозвище

Ассоциативные

- Люди_участвующие_события
 - человек_ид
 - событие_ид
- Качества_человека
 - человек_ид
 - качество_ид
- Эмоции_человека
 - человек_ид
 - эмоция_ид
- Отношения_людей_друг_к_другу
 - человек_1_ид
 - человек_2_ид
 - тип_взаимоотношений
 - начало_взаимоотношений
 - конец_взаимоотношений

Описание бизнес-процессов

Доктор Маруки, обретя свои силы, решает изменить этот мир к лучшему. Ему требуется создать новый мир, но для этого нужно, чтобы все люди были счастливы. Проблема в том, что у всех разное понимание счастья. У доктора своя философия на этот счет: счастье в его случае — это когда не наносится какой-либо вред другому. Поэтому свое дело он начнет с корректировки умов людей.

1. Изменение восприятия человека

- a. Доктор запрашивает нужного человека
- b. Находится необходимый человек в Реальном мире
- c. По нему находится его Тень
- d. Смотрится его Тень и проверяется ее желание
 - i. Если ее желание противоречиво доктору (исчезновение/ухудшение качеств, появление/исчезновение событий), то его желание изменяется и обрastaет положительными чертами (появление новых характеристик, изменение плохих событий).
 - ii. Если желание положительное для доктора, то оно никак изменяется.
- e. Доктор получает ответ в виде измененное желания запрашиваемого человека (или неизменного, если с ним все в порядке). То есть если этот человек желал кому-то изменение качеств на отрицательные по мнению Доктора (человек хотел сделать кого-то Гордым или наоборот, отнять его добрые черты, такие как Милосердие), то Маруки получит от системы желание, в котором видно, что человек хочет обрести такую черту, как Милосердие.

P.S. Доктору не обязательно знать, какой именно был человек, потому что этот процесс изменения мира делает его Персона, поэтому ему нет разницы, какой вернется человек. Главное — чтоб его желание было положительным.

Научившись менять желания людей, доктор пошел дальше. Теперь он готов воплощать их желание в жизнь.

2. Исполнение желаний

- a. Процесс аналогичный изменению восприятия человека, но Доктору ничего не возвращается
- b. Берется Желание человека и сопоставляется с самим человеком, его событиями, со связанными с ним людьми.
- c. При нахождении расхождений Желания с реальным миром, мир изменяется под стать Желанию
- d. Доктору ничего не возвращается
- e. В Реальном мире изменяются параметры Людей (включая запрашиваемого человека), соответствующие его желанию. То есть если человек хотел, чтобы у него у него появилась такое качество, как Смелость, то он его обретае. Аналогично и меняются другие люди: если кто-то из них был мертв, его дата смерти попросту стирается.

Исполнив несколько желаний, Маруки решает изменить весь мир, сделав всех людей на планете счастливыми.

3. Создание лучшего мира

- a. Стоит уточнить, что способности Доктора по правилам самой игры заключаются в том, что он именно объединяет оба мира в один целый, который он сможет корректировать и контролировать; создать утопию для всех людей, где все счастливы.
- b. Итак, что же происходит в нашей системе:
 - i. Начинается процесс аналогичный Исполнению желаний человека, НО он применяется ко ВСЕМ людям в Реальном мире
 - ii. После таких изменений в системе все мечты людей скорректированы и исполнены, у них больше нет никаких страхов и переживаний, поэтому и существование Метаверса больше не требуется
 - iii. Способности Доктора заключаются в объединении миров, потому что с помощью Метаверса он способен менять и исполнять желания людей, поэтому начинается процесс объединения
 - iv. Тень являются отражением негативных эмоций человека, но если все люди счастливы, то соответственно их существование более не требуется и от них попросту мы избавимся, поэтому никак их не трогаем
 - v. Затем в самом Метаверсе есть свои Локации, отражающие реальный мир. Поэтому Объекты из Метаверса переносятся в Реальный мир (в Локациях появляются Объекты как Достопримечательности).
 - vi. Т.к. теней больше нет, а миры объединились, то Метаверс попросту уничтожается, а Реальный мир становится единственным миром в системе
 - vii. Доктору ничего не возвращается

После завершения все, что может Доктор, это пожимать плоды своего творения. Поэтому ему остаётся просто наблюдать за происходящим

4. Проверка человека

- a. Доктор запрашивает конкретного человека
- b. Ему возвращается Человек

Еще можно добавить Исполнение желания без предварительного изменения восприятия человека, процесс аналогичный, но может вернуться ошибка, что желание не исполнится из-за противоречий.

Федеральное государственное автономное
образовательное учреждение высшего образования

Университет ИТМО

**Курсовая работа по дисциплине
"Информационные системы и базы данных".**

Этап 2

Выполнили:

Марков Петр Денисович
Кривоносов Егор Дмитриевич

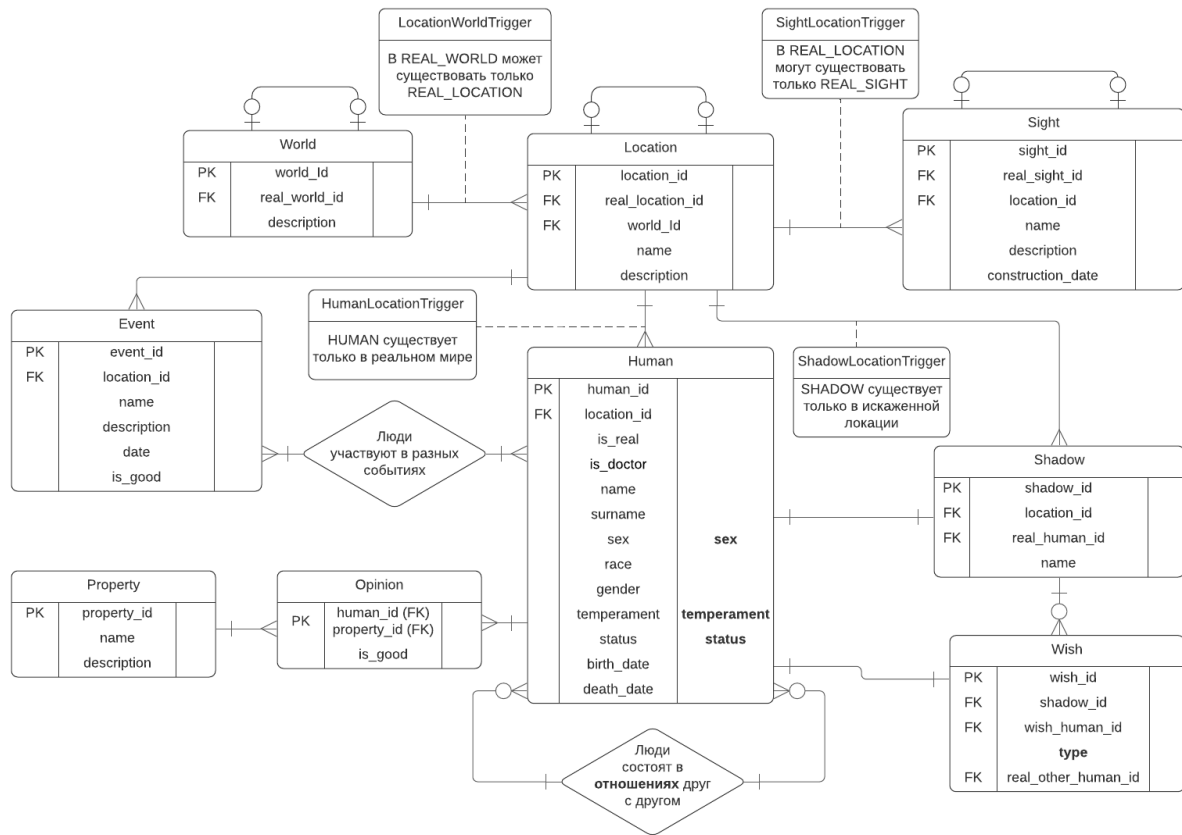
Группа: Р33111

Преподаватель:

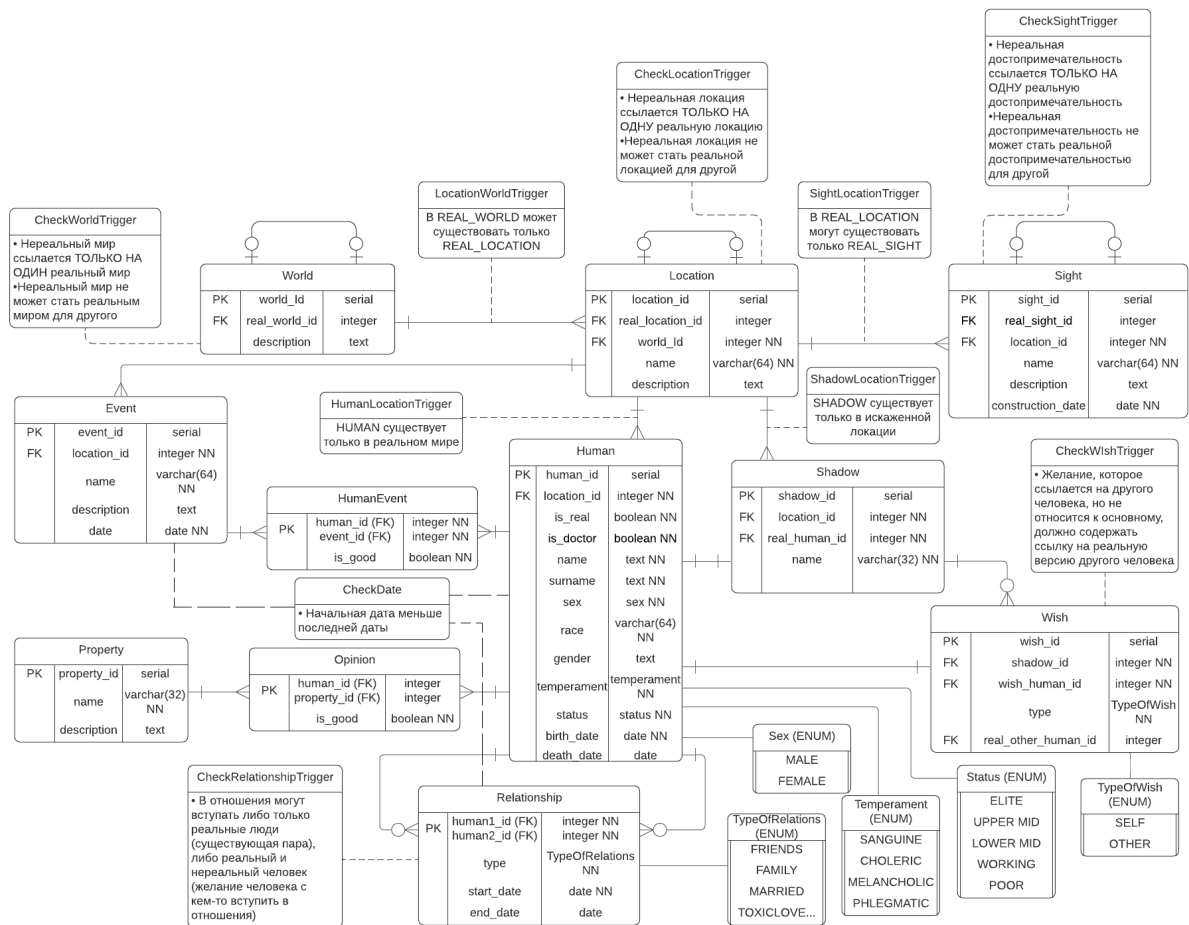
Николаев Владимир Вячеславович

2021 г.

Инфологическая модель



Даталогическая модель



Федеральное государственное автономное
образовательное учреждение высшего образования

Университет ИТМО

**Курсовая работа по дисциплине
"Информационные системы и базы данных".**

Этап 3

Выполнили:

Марков Петр Денисович
Кривоносов Егор Дмитриевич

Группа: Р33111

Преподаватель:

Николаев Владимир Вячеславович

2021 г.

Этап 3

Реализовать даталогическую модель в реляционной СУБД PostgreSQL:

- Создать необходимые объекты базы данных.
- Заполнить созданные таблицы тестовыми данными.
- Сделать скрипты для:
 - создания/удаления объектов базы данных;
 - заполнения/удаления созданных таблиц.
- Обеспечить целостность данных при помощи средств языка DDL.
- Добавить в базу данных триггеры для обеспечения комплексных ограничений целостности.
- Реализовать функции и процедуры на основе описания бизнес-процессов (из этапа №1).
- Произвести анализ использования созданной базы данных:
 - выявить наиболее часто используемые запросы к объектам базы данных;
 - результаты представить в виде текстового описания.
- Создать индексы и доказать, что они полезны для вашей базы данных:
 - доказательство должно быть приведено в виде текстового описания.

Создание и заполнение базы данных данными

В ходе выполнения этапа 3.1 были созданы следующие скрипты:

- [create.sql](#) (создает базу данных)
- [data.sql](#) и [mega.sql](#) (заполняет базу данных данными)
- [delete.sql](#) (удаляет данные из базы данных)
- [drop.sql](#) (удаляет таблицы базы данных)

Дополнительные скрипты:

- [triggers.sql](#) (создает триггеры)
- [functions.sql](#) (создает функции)
- [indexes.sql](#) (создает индексы)

Описание

Перед нами стояли задачи по реализации различных функций и триггеров для нашей базы данных. Наша основная таблица - human. Функционал пользователя подразумевает взаимодействие с этой таблицей, а также связанные с ней таблицы. В первую очередь наш пользователь сможет “изменять желания людей”, а для этого ему нужен только human, но и его желания, которые можно получить из связанной таблицы shadow. Кроме того, у людей есть разные отношения к происходящим в мире Persona событиям (humanevent) и у них есть свои собственные характеристики, которые они способны оценивать (opinion). Также, сами люди способны могут вступать в отношения (relationship) или мечтают о них. Основная идея при создании базы данных заключалась в том, что у нас есть нереальные версии людей, которые подразумевают их улучшенные/желаемые версии. При чем это могут быть как желаемые версии их самих, так и желаемые версии других людей. Но по логике Persona свобода одного человека заканчивается там, где начинается свобода другого, поэтому это ставит

некоторые ограничения на функции. Также, у нас не могут существовать нереальные объекты, если не существуют их реальные версии, что отразилось в триггерах. А также логика взаимодействия реальных людей и их нереальных версий отражена в триггерах.

В работе всех функций мы не используем почти различные соединения таблиц (left / right join), только при получении дополнительной информации о людях (inner join). В основном мы используем подзапросы, потому что в данном случае они будут эффективнее работать, чем делать через join'ы (пример приведен в таблице ниже). Нам не нужно получать из других таблиц все данные, а только, некоторые поля (большая часть запросов у нас связана с human_id), которые мы указываем через проекцию, тем самым мы экономим память.

На примере ниже мы видим 2 аналогичных запроса, которые выдают нереальную версию по id реального человека. Их отличия различаются только в реализации: с INNER JOIN и подзапросом. Мы выяснили, что с подзапросом запрос работает быстрее. Это происходит за счет того, что подзапрос возвращает только один объект, а затем по нему фильтруем и находим подходящий, а также не тратится время на создание дополнительной таблицы хешей.

```
EXPLAIN ANALYSE SELECT wish_human_id FROM wish
JOIN shadow s on wish.shadow_id = s.shadow_id
WHERE wish.type = 'SELF' AND s.real_human_id = 1111
```

Output × Result 1 ×

10 rows

QUERY PLAN

1	Hash Join (cost=8.03..41.41 rows=1 width=4)...
2	Hash Cond: (wish.shadow_id = s.shadow_id)
3	-> Seq Scan on wish (cost=0.00..27.75 ro...
4	Filter: (type = 'SELF'::typeofwish)
5	-> Hash (cost=8.02..8.02 rows=1 width=4)...
6	Buckets: 1024 Batches: 1 Memory Us...
7	-> Index Scan using real_human_id_i...
8	Index Cond: (real_human_id = 1...
9	Planning time: 2.253 ms
10	Execution time: 1.490 ms

```
EXPLAIN ANALYSE SELECT wish_human_id FROM wish
WHERE wish.type = 'SELF' AND wish.shadow_id =
(
    SELECT shadow_id FROM shadow
    WHERE shadow.real_human_id = 1111
);
```

Output × Result 2 ×

8 rows

QUERY PLAN

```
1 Seq Scan on wish (cost=8.02..39.52 rows=...
2   Filter: ((type = 'SELF'::typeofwish) AN...
3   Rows Removed by Filter: 1499
4   InitPlan 1 (returns $0)
5     -> Index Scan using real_human_id_id...
6         Index Cond: (real_human_id = 11...
7 Planning time: 0.213 ms
8 Execution time: 0.334 ms
```

Триггеры

В ходе выполнения этапа 3.2 были реализованы следующие триггеры:

- **LOCATION_INSERT** - проверяет, что на реальную локацию не ссылается несколько искаженных; проверяет, что искаженная локация не может быть реальной для другой искаженной локации.
- **SIGHT_INSERT** - проверяет, что на реальная достопримечательность не ссылается несколько искаженных; проверяет, что искаженная достопримечательность не может быть реальной для другой искаженной достопримечательности.
- **WORLD_INSERT** - проверяет, что на реальный мир не ссылается несколько искаженных; проверяет, что искаженный мир не может быть реальным для другого искаженного мира. Проверяет, что мира существует только 2 (искаженный и реальный)
- **ShadowHumanTrigger** - проверяет, что у 1 человека, может быть только 1 тень.
- **LocationWorldTrigger** - проверяет, что реальная/искаженная локация существует в реальном/искаженном мире. Локация является реальной, если у нее поле `real_location_id` является NULL

- **SightLocationTrigger** - проверяет, что реальная/искаженная достопримечательность существует в реальной/искаженной локации. Достопримечательность является реальной, если у ней поле *real_sight_id* является NULL.
- **HumanLocationTrigger** - проверяет, что человек существует в реальной локации. Локация является реальной, если у нее поле *real_location_id* является NULL.
- **ShadowLocationTrigger** - проверяет, что тень человека существует в искаженной локации. Локация является искаженной, если у нее поле *real_location_id* не является NULL.
- **TypeOfWishTrigger** - проверяет, что если желание ссылается на другого человека, то поле *real_other_human_id* не должно быть NULL (другой человек должен существовать в реальном мире).
- **RelationshipTrigger** - проверяет, что отношения могут быть только у двух реальных людей, или у реального и нереального человека. Ситуации когда они оба нереальные - не может быть. Проверяет, что отношения между реальным и не реальным самим собой не может быть.

Функции

В ходе выполнения этапа 3.2 были реализованы следующие функции на основе бизнес-процессов (из этапа №1):

- **IS_REAL_HUMAN** - возвращает is_real человека (настоящий ли человек или вымышленный), участвует во многих проверках
- **GET_PERSON_IMAGINE_SELF_ID** - возвращает мечту самого человека (его вымышленная версия), нужен в триггере
- **GET_NUM_OF_STATUS** - возвращает номер статуса по приоритету, нужен, чтобы сравнивать благосостояние человека (от 'POOR' до 'ELITE')
- **IS_PROPERTY_GOOD_FOR_DOCTOR** - возвращает is_good определенной характеристики, но по версии самого доктора, нужна для проверок
- **IS_EVENT_GOOD_FOR_DOCTOR** - возвращает is_good определенного события, но по версии самого доктора, нужна для проверок
- **IS_DOCTOR** - возвращает is_doctor человека (является ли он доктором), нужна для проверок и начала действий
- **CHANGE_HUMAN_WISHES** - меняет желание человека так, чтобы: 1) его статус не стал ниже того, что был; 2) удаляет плохие по мнению доктора мнения; 3) меняет мнение людей на событие на положительное, если так считает доктор
- **MAKE_WISHES_COME_TRUE** - исполняет желание человека так, чтобы: 1) параметры человека стали такими, какими он желал; 2) воскрешает мертвого человека (убирает дату смерти); 3) если 2 человека хотят вступить в отношения друг с другом, то у них это получается; 4) стирает из истории плохое по мнению доктора событие; 5) удаляет выполненное желание
- **MAKE_ALL_WISHES_COME_TRUE** - выполняет все желания и объединяет миры

Желтое - те функции, которые мы решили оставить реализацию на 4 этапы курсовой работы, потому что на наш взгляд они получаются очень объемные (по памяти) и

эффективнее и быстрее они будут исполняться (обрабатываться) на серверной части нашего сайта за счет того, что тогда нам не придется делать многократное кол-во одних и тех же запросов и проверок.

Остальные функции мы оставили, потому что с их помощью будет удобно и быстро реализовать бизнес-процессы, а также они применяются в триггерах.

Анализ использования созданной базы данных

В нашей базе данных основная структура - Human. Human у нас как хранит настоящих людей, так и их вымышленные версии - их мечты о самих себе. Поэтому получение human идет постоянно и во всех запросах. Кроме того, у каждого реального человека есть своя тень - Shadow, у которой есть мечты - Wish. Также в мире существуют определенные события - Event, а также характеристики людей - Property. Но они не так важны. Важно именно то, как люди к ним относятся. Поэтому в opinion и humanevent у нас довольно много строк, потому что сколько людей - столько и мнений.

Бизнес логика у нас завязана на получении human и его изменении, а также всех ассоциативных связей, где он участвует. Наша реализация позволяет это делать без лишних проблем, а структура БД позволяет нам хранить разные версии людей в одной таблице, что упрощает нам сравнение и проверки при исполнении желаний.

Ассоциативные связи (например, relationship) не только связывают объекты в базе данных, но и хранят дополнительные параметры (время создания, тип отношений, отношения человека к этой характеристике). Такой подход очень удобный, он дает нам гибкость при работе с мнениями людей и позволяет нам проверять и изменять желания людей с помощью мнений людей.

Созданные нами функции помогут нам получать нужных human и их параметры, особенно is_real. Кроме того, функции используются в некоторых триггерах, что довольно удобно, т.к. нам не требуется по новой писать ту же логику.

Индексы

В ходе выполнения этапа 3.2 были проанализированы функции и выявлены возможные индексы:

Были созданы следующие HASH индексы:

- Таблица Human:
 - human_id_idx - для нахождения человека по нужному ид.
 - true_idx - для определения является ли человек реальным.
- Таблица Shadow:
 - shadow_id_idx - для нахождения тени по нужному ид.
 - real_human_id_idx - для нахождения тени для определенного человека.
- Таблица Wish:

- real_other_human_id_idx - для получение человека, с которым связано желание другого человека.
 - type_idx - для нахождения желаний с определенным типом.
- Таблица HumanEvent:
 - event_id_idx - для определения и обновления нужного нам события.
- Таблица Opinion:
 - human_opinion_id_idx - для определения качества определенного человека.

Доказательство

Так как, большая часть наших функций использует в основном обращение к большим таблицам Human, Shadow, Wish, Event и поиск по айдишникам в них, поэтому при оптимизации нужно делать упор на них и создать для них индексы.

При создании запросов, мы используем только простые условия равенства (=) без каких-либо операторов сравнения (>, <, <=, >=). Поэтому логичнее будет создавать hash индексы. Они в данном случае работают быстрее чем те же самые btree и **занимают меньше памяти**. Но так как есть одна особенность, **когда мы создаем РК столбцы они изначально имеют btree индекс**. Поэтому скорость выполнения одного запроса и разница между hash и btree не значительна, но если его выполнять в цикле, то разница достаточно сильно видна (Даже воспользовавшись explain analyze видно, что в место btree индекса, у нас выбирается нами созданный hash индекс так как он эффективнее).

Ниже представлены примеры EXPLAIN ANALYZE насколько hash индексы ускоряют работу запросов:

SELECT * FROM human WHERE human_id = 89;

До создания индексов:

```

QUERY PLAN
1  Index Scan using human_pkey on human  (cost=0.28..8.30 rows=1 width=70) (actual time=0.034..0.036 rows=1 loops=1)
2    Index Cond: (human_id = 1242)
3  Planning time: 0.145 ms
4  Execution time: 0.181 ms

```

После создания индексов:

```

QUERY PLAN
1  Index Scan using human_id_idx on human  (cost=0.00..8.02 rows=1 width=70) (actual time=0.087..0.088 rows=1 loops=1)
2    Index Cond: (human_id = 1242)
3  Planning time: 0.268 ms
4  Execution time: 0.110 ms

```

Описание:

Разница времени выполнения: 1.65 раза. Все из-за того, что hash индекс быстрее производит поиск используя оператор "=" чем изначально созданный для РК индекс btree. Как и писалось выше у нас нигде не используются другие операторы сравнения и из-за это hash индекс является эффективнее в нашем случае.


```
SELECT * FROM human
JOIN shadow s ON human.human_id = s.real_human_id
JOIN wish w ON s.shadow_id = w.shadow_id
WHERE human_id = 1321;
```

До создания индексов:

```
QUERY PLAN
1 Nested Loop (cost=8.59..39.41 rows=1 width=108) (actual time=0.135..0.270 rows=1 loops=1)
2   -> Hash Join (cost=8.31..31.10 rows=1 width=38) (actual time=0.129..0.262 rows=1 loops=1)
3       Hash Cond: (w.shadow_id = s.shadow_id)
4       -> Seq Scan on wish w (cost=0.00..19.50 rows=1250 width=20) (actual time=0.010..0.062 rows=1250 loops=1)
5       -> Hash (cost=8.29..8.29 rows=1 width=18) (actual time=0.009..0.009 rows=1 loops=1)
6           Buckets: 1024 Batches: 1 Memory Usage: 9kB
7           -> Index Scan using shadow_real_human_id_key on shadow s (cost=0.28..8.29 rows=1 width=18) (actual time=0.006..0.006 rows=1 loops=1)
8               Index Cond: (real_human_id = 1321)
9       -> Index Scan using human_pkey on human (cost=0.28..8.30 rows=1 width=70) (actual time=0.005..0.006 rows=1 loops=1)
10          Index Cond: (human_id = 1321)
11 Planning Time: 0.205 ms
12 Execution Time: 0.295 ms
```

После создания индексов:

```
QUERY PLAN
1 Nested Loop (cost=0.00..24.07 rows=1 width=108) (actual time=0.017..0.019 rows=1 loops=1)
2   -> Nested Loop (cost=0.00..16.05 rows=1 width=88) (actual time=0.013..0.014 rows=1 loops=1)
3       -> Index Scan using human_id_idx on human (cost=0.00..8.02 rows=1 width=70) (actual time=0.007..0.008 rows=1 loops=1)
4           Index Cond: (human_id = 1321)
5       -> Index Scan using real_human_id_idx on shadow s (cost=0.00..8.02 rows=1 width=18) (actual time=0.005..0.005 rows=1 loops=1)
6           Index Cond: (real_human_id = 1321)
7   -> Index Scan using shadow_type_idx on wish w (cost=0.00..8.02 rows=1 width=20) (actual time=0.003..0.004 rows=1 loops=1)
8       Index Cond: (shadow_id = s.shadow_id)
9 Planning Time: 2.581 ms
10 Execution Time: 0.045 ms
```

Описание:

В первом случае наш запрос использует соединение вложенными циклами и хэш соединение из-за того, что таблицы имеют большой объем данных. Когда происходит хэш соединение, СУБД тратит дополнительное время на создание таблицы хэш индексов.

Во втором же случае у нас используется два соединения вложенными циклами и не тратится дополнительное время на создание хэш таблицы, потому что они уже создались, когда мы создали hash индексы.

Разница времени выполнения: 6.56 раз.

```
SELECT CHANGE_HUMAN_WISHES(217, 1);
```

До создания индексов:

```
QUERY PLAN
1 ProjectSet (cost=0.00..5.27 rows=1000 width=32) (actual time=2.966..2.967 rows=1 loops=1)
2   -> Result (cost=0.00..0.01 rows=1 width=0) (actual time=0.001..0.001 rows=1 loops=1)
3 Planning Time: 0.014 ms
4 Execution Time: 2.984 ms
```

После создания индексов:

QUERY PLAN	
1	ProjectSet (cost=0.00..5.27 rows=1000 width=32) (actual time=0.869..0.870 rows=1 loops=1)
2	-> Result (cost=0.00..0.01 rows=1 width=0) (actual time=0.001..0.001 rows=1 loops=1)
3	Planning Time: 0.033 ms
4	Execution Time: 0.895 ms

Описание:

Разница времени выполнения: 3.34 раза.

Федеральное государственное автономное
образовательное учреждение высшего образования

Университет ИТМО

**Курсовая работа по дисциплине
"Информационные системы и базы данных".**

Этап 4

Выполнили:

Марков Петр Денисович
Кривоносов Егор Дмитриевич

Группа: Р33111

Преподаватель:

Николаев Владимир Вячеславович

2021 г.

Username

Password

Sign Up

Doctor's Parameters

Name

Surname

Sex

- Select Sex -

Race

Gender

Temperament

- Select Temperament -

Status

- Select Status -

Save

Network Error

My Profile

Name

Write your name

Surname

Write your surname

Sex

- Select Sex -

Race

Write your Race

Gender

Write your Gender

Temperament

- Select Temperam

Status

- Select Status -

Add doctor info

Network Error

#	Name	Surname	Location_id	Real?	Doctor?	Sex	Race	Gender	Temperament	Status	Birth_date	Death_date
1	Egor	Krivososov	1	Да	Да	MALE	GREEN	wtf	SANGUINE	ELITE	17.10.2001	-
2	Petr	Markov	1	Нет	Да	MALE	PDF MAN	wtf	SANGUINE	ELITE	24.02.2002	-

PreviousNext

Network Error

#	Name	Surname	Location_id	Real?	Doctor?	Sex	Race	Gender	Temperament	Status	Birth_date	Death_date							
1	Egor	Krivososov	1	Да	Да	MALE	GREEN	wtf	SANGUINE	ELITE	17.10.2001	-							
Relationships																			
Name		Surname	Type	Start_date	End_date														
Petr		Markov	FRIENDS	2019	X3														
Events																			
#	Location_id		Name	Description	Date														
1	1		Написание курсача	Может не надо???	01.09.2021														
2	1		CS:GO	Сюрфит на рекордики	14.01.2022														
Property																			
#	Name		Description																
1			Дружелюбный																
2			Не дружелюбный																
3			Секси																
2	Petr	Markov	1	Нет	Да	MALE	PDF MAN	wtf	SANGUINE	ELITE	24.02.2002	-							
Relationships																			
Name		Surname	Type	Start_date	End_date														
Egor		Krivososov	FRIENDS	2019	X3														

PreviousNext

#	Name	Surname	Location_id	Real?	Doctor?	Sex	Race	Gender	Temperament	Status	Birth_date	Death_date
1	Egor	Krivososov	1	Да	Да	MALE	GREEN	wtf	SANGUINE	ELITE	17.10.2001	-
Relationships												
Events												
#	Location_id		Name	Description		Date		Good?				
1	1		Написание курсача	Может не надо???		01.09.2021		<div>YesNo--</div>				
2	1		CS:GO	Сюрфит на рекордики		14.01.2022		<div>YesNo--</div>				
Property												
2	Petr	Markov	1	Нет	Да	MALE	PDF MAN	wtf	SANGUINE	ELITE	24.02.2002	-

скриншоты + описание бизнес логики

TODO - Egor (Frontend)

- ~~1. Разобраться как работает проект :)~~
- ~~2. Поправить форму регистрации и входа (home страницы нет, если не вошёл).~~
- ~~3. Добавить страницу для регистрации доктора (параметры).~~
- ~~4. Добавить загрузку таблицы при удачном входе в систему.~~
- ~~5. В Figma набросать концепт основной таблицы с кнопками.~~
- 6. Как будет готово с бэком добавить еще 1-2 параметра к людям в таблицу.**
- ~~7. Попробовать пофиксить таблицу под мобилки.~~
- ~~8. Добавить кнопки good в event и property~~
- ~~9. Настройка профиля доктора у пользователя~~
- ~~10. Добавить дату рождения в профиль и регистрацию.~~

Вопросики

- Подгрузку строк таблицы будем делать через бэк? Чтобы на фронт сразу всю таблицу не отсылать?

ОТВЕТ: Да, по 10 на страницу

- Как идея сделать кнопки при нажатии на любого человека? [Пример](#)

ОТВЕТ: Лучше сделай развертку других параметров при нажатии на человека

- И смесь 2 таблиц: [Таблица 1](#) + [Таблица 2](#)

ОТВЕТ: Да

Новое:

~~1) В профиль добавить справа таблицы с Event и Property с кнопками для выбора isGood~~

~~2) В главной таблице добавить 2 кнопки при нажатии на человека~~

										апреля 2000 г.		
Relationships												
Property												
2	HRNBNEMS	6D8FYK1S	Miura	Her	Her	MALE	European	Heterosexual	CHOLERIC	UPPER	суббота, 1 апреля 2000 г.	-
3	1E6SB5WQ	ZCSLHISF	Miura	Да	Her	MALE	European	Heterosexual	PHLEGOMATIC	WORKING	суббота, 1 апреля 2000	-

~~3) В главной таблице сверху сделать кнопку тип MAGIC~~

```
Home.vue:57da:289
▼ (8) [{...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}] ⓘ
  ▼ 0:
    birthDate: "2000-04-01T00:00:00"
    deathDate: null
    gender: "Heterosexual"
    ▶ humanEventResponseDtoList: []
    id: 1
    isDoctor: false
    isReal: true
    locationName: "Miura "
    name: "HRNBNEMS"
    ▶ opinionResponseDtoList: (2) [{...}, {...}]
    race: "European"
    ▼ relationshipResponseDtoList: Array(2)
      ▼ 0:
        endDate: null
        name: "10N1Y16W"
        startDate: "2020-06-13T00:00:00"
        surname: "939G9Y50"
        type: "MARRIED"
        ▶ [[Prototype]]: Object
      ▶ 1: {name: 'X082XZP3', surname: '13FBJBAL', ...}
        length: 2
        ▶ [[Prototype]]: Array(0)
        sex: "MALE"
        status: "WORKING"
        surname: "6D8FYK1S"
        temperament: "PHLEGMATIC"
        ▶ [[Prototype]]: Object
```

```
isReal: true
locationName: "Miura "
name: "HRNBNEMS"
▼ opinionResponseDtoList: Array(2)
  ▼ 0:
    description: "The quality of being loyal"
    id: 3
    isGood: false
    name: "Loyalty"
    ▶ [[Prototype]]: Object
  ▶ 1: {id: 5, name: 'Kindness', description: '...' }
    length: 2
    ▶ [[Prototype]]: Array(0)
  race: "European"
▼ relationshipResponseDtoList: Array(2)
  ▼ 0:
    endDate: null
```

```

4: {id: 5, name: 'X082XZP3', surname: '13FB3BA
5: {id: 6, name: 'X082XZP3', surname: '13FB3BA
6:
  birthDate: "2000-04-01T00:00:00"
  deathDate: null
  gender: "Heterosexual"
  humanEventResponseDtoList: Array(1)
    ▼ 0:
      date: "2014-03-08T00:00:00"
      description: "International Women's Day"
      id: 5
      isGood: false
      locationName: "Miura "
      name: "Woman Day"
      ▶ [[Prototype]]: Object
      length: 1
      ▶ [[Prototype]]: Array(0)
    id: 7
    isDoctor: false

```

CHANGE WISHES / REALIZE WISHES

TODO - Петя (Backend)

1. `SELECT setval('human_human_id_seq', 3000, true);`
2. Добавь функционал создания полей доктора
3. Добавь функционал добавления/изменения пропертей и хуман ивентов
4. Добавь основные 3 функции
5. Проверь чо там с функциями и где их юзать
6. Соедини с фронтом
7. Готово!