Экзаменационный проект по дисциплине Базы данных

студента гр. **R3495**

**Гороховой Надежды**

по теме

**Деревня ниндзя**

# Описание проекта

В деревне ниндзя живут в основном ниндзя, встречаются иногда простые жители. Ниндзя имеют разные звания, можно провести аналогию военными званиями. Ниндзя владеет различными техниками (technique). Иногда, чтобы владеть техникой A, требуется владеть техниками B и C, например.

В данном мире есть биджу (Bidju) - огромные сгустки энергии, которые имеют материальное воплощение в виде зверей. Они могут быть помещены (запечатаны) в ниндзя. Ниндзя в котором запечатан биджу называется джинчуурики (Jinchuuriki).

# Построение отношений

В результате предварительного проектирования были выделены следующие отношения:

* NinjaVillage - деревня ниндзя
* Villager - житель деревни. Житель может относится лишь к одной деревне.
* Ninja - ниндзя. Ниндзями не рождаются, а становятся жители деревни.
* Clan - клан или группа ниндзя связанная родством, как правило. Аналогия с реальным миром - знатный род. Все ниндзя состоящие в клане должны иметь фамилию совпадающую с названием клана.
* Bidju - биджу. Это огромные сгустки энергии, которые имеют материальное воплощение в виде хвостатых зверей. Количество хвостов прямопропорционально определяют силу этого зверя. Их не так много, поэтому имена уникальные, количество хвостов так же уникально. Не бывает биджу с одинаковым количеством хвостов.
* Jinchuuriki (джинчуурики). Биджу может быть запечатан в ниндзя, тогда ниндзя является джинчуурики этого биджу. Биджу может быть запечатан одновременно лишь в одного ниндзя, и в джинчуурики одновременно может быть запечатан лишь один биджу. На протяжении всей своей жизни ниндзя может быть джинчуурики не более одного биджу, потому что “изъятие” биджу из ниндзя заканчивается смертью для последнего. Но биджу в разные временные периоды может быть запечатан в разных ниндзя.
* Technique - техники (приёмы), которыми могут владеть ниндзя.
* TechniqueRequirement - для некоторых техник может потребоваться владение другой (другими) техниками (техникой).
* Skill - “ниндзя владеет техникой”

**Замечание 1**. Для доказательства, что отношение находится в НФБК требуется показать, что в каждой нетривиальной фз этого отношения левая часть - надключ. Я привожу фз не все, а основные. Сейчас напишу тут, чтобы не повторяться:

Чтобы показать, что отношение находится в НФБК, надо рассмотреть все нетривиальные фз и доказать, что левая часть - надключ. Если есть разные фз вида X -> Y и X - надключ, то надо доказать, что в фз X’ -> Y’ полученным по правилам вывода X’ - тоже надключ. По правилу дополнения можно получить фз, где левая часть будет отличаться, но если X -> Y и X - надключ, то и очевидно, что X Z будет являться надключом.

## Отношение NinjaVillage

Атрибуты:

* id - суррогатный ключ
* name - уникальноназвание деревни
* description - описание
* foundationDate - дата основания
* villagerId - внешний ключ отношения Villager для соблюдения ограничения, что в деревни проживал хотя бы один человек. Для удобства будет хранится id основателя деревни.

Функциональные зависимости:

* id -> name
* id -> description
* id -> foundationDate
* id -> villagerId
* name -> id
* name -> description
* name -> foundationDate
* name -> villagerId
* villagerId -> id
* villagerId -> name
* villagerId -> descriptionId
* villagerId -> foundationDate

\* Остальные фз могут быть получены из правил вывода + тривиальные фз

Нормализация:

* Нет повторяющихся групп атрибутов, все атрибуты атомарны, есть ключ (3 ключа: id, name, villagerId, так как замыкание атрибутов каждого из этих атрибутов совпадает с множеством всех атрибутов) => находится в 1НФ
* Находится в 1НФ, нет составных ключей => находится в 2НФ
* Находится в 2НФ, все неключевые атрибуты зависят напрямую от ключей => 3НФ
* Во всех ФЗ вида X -> Y , X - надключ (рассмотрено в 1м пункте), и во всех нетривиальных зависимостях полученных по правилам вывода левая часть тоже надключ (см. Замечание 1) => находится в НФБК
* По второй теореме Дейта-Фейгина находится в 4НФ
* По первой теореме Дейта-Фейгина находится в 5НФ

## Отношение Villager

Атрибуты:

* id - суррогатный ключ
* firstName - имя жителя
* lastName - фамилия жителя
* birthday - дата рождения жителя
* deathDate - дата смерти жителя
* villageId - внешний ключ деревни, в которой живет житель

Функциональные зависимости:

* id -> name
* id -> firstName
* id -> lastName
* id -> birthday
* id -> deathDate
* id -> villageId

\* Остальные фз могут быть получены из правил вывода + тривиальные фз

Нормализация:

* Нет повторяющихся групп атрибутов, все атрибуты атомарны, есть ключ (id - ключ, так как замыкание атрибутов совпадает с множеством всех атрибутов) => находится в 1НФ
* Находится в 1НФ, нет составных ключей => находится в 2НФ
* Находится в 2НФ, все неключевые атрибуты зависят напрямую от ключей => 3НФ
* Во всех ФЗ вида X -> Y , X - надключ (рассмотрено в 1м пункте), и во всех нетривиальных зависимостях полученных по правилам вывода левая часть тоже надключ (см. Замечание 1) => находится в НФБК
* По второй теореме Дейта-Фейгина находится в 4НФ
* По первой теореме Дейта-Фейгина находится в 5НФ

## Отношение Ninja

Атрибуты:

* ninjaId - внешний ключ жителя деревни, к которому привязана эта запись, одновременно и основной ключ
* status - статус ниндзя: активен, на пенсии, мертв.
* rank - ранг ниндзя
* clanId - внешний ключ клана, в котором состоит ниндзя

Функциональные зависимости:

* ninjaId -> status
* ninjaId -> rank
* ninjaId -> clanId

\* Остальные фз могут быть получены из правил вывода + тривиальные фз.

Нормализация:

* Нет повторяющихся групп атрибутов, все атрибуты атомарны, есть ключ (ninjaId - ключ, потому что замыкание атрибутов совпадает с множеством всех атрибутов) => находится в 1НФ
* Находится в 1НФ, нет составных ключей => находится в 2НФ
* Находится в 2НФ, все неключевые атрибуты зависят напрямую от ключей => 3НФ
* Во всех ФЗ вида X -> Y , X - надключ (рассмотрено в 1м пункте), и во всех нетривиальных зависимостях полученных по правилам вывода левая часть тоже надключ (см. Замечание 1) => находится в НФБК
* По второй теореме Дейта-Фейгина находится в 4НФ
* По первой теореме Дейта-Фейгина находится в 5НФ

## Отношение Bidju

Атрибуты:

* id - суррогатный ключ
* name - уникальное имя биджу
* tailsAmount - количество хвостов, уникально для каждого биджу.
* description - описание биджу

Функциональные зависимости:

* id -> name
* id -> tailsAmount
* id -> description
* name -> id
* name -> tailsAmount
* name -> description
* tailsAmount -> id
* tailsAmount -> name
* tailsAmount -> description

\* Остальные фз могут быть получены из правил вывода + тривиальные фз

Нормализация:

* Нет повторяющихся групп атрибутов, все атрибуты атомарны, есть ключ (3 ключа: id, name, tailsAmount, так как замыкание атрибутов каждого из этих атрибутов совпадает с множеством всех атрибутов) => находится в 1НФ
* Находится в 1НФ, нет составных ключей => находится в 2НФ
* Находится в 2НФ, все неключевые атрибуты зависят напрямую от ключей => 3НФ
* Во всех ФЗ вида X -> Y , X - надключ (рассмотрено в 1м пункте), и во всех нетривиальных зависимостях полученных по правилам вывода левая часть тоже надключ (см. Замечание 1) => находится в НФБК
* По второй теореме Дейта-Фейгина находится в 4НФ
* По первой теореме Дейта-Фейгина находится в 5НФ

## Отношение Jinchuuriki

Атрибуты:

* ninjaId - внешний ключ, идентификатор ниндзя, который является джинчуурики. Ниндзя может быть джинчуурики лишь один раз, поэтому этот атрибут является ключом.
* bidjuId - внешний ключ, идентификатор биджу, который запечатан в джинчуурики
* fromDate - когда биджу был запечатан
* toDate - когда биджу был освобожден из тела джинчуурики, обычно такое происходит в результате смерти последнего

Функциональные зависимости:

* ninjaId -> bidjuId
* ninjaId -> fromDate
* ninjaId -> toDate

\* Остальные фз могут быть получены из правил вывода + тривиальные фз

Нормализация:

* Нет повторяющихся групп атрибутов, все атрибуты атомарны, есть ключ (ninjaId - ключ, потому что замыкание атрибутов совпадает с множеством всех атрибутов) => находится в 1НФ
* Находится в 1НФ, нет составных ключей => находится в 2НФ
* Находится в 2НФ, все неключевые атрибуты зависят напрямую от ключей => 3НФ
* Во всех нетривиальных ФЗ вида X -> Y , X - надключ (рассмотрено в 1м пункте), и во всех нетривиальных зависимостях полученных по правилам вывода левая часть тоже надключ (см. Замечание 1) => находится в НФБК
* По второй теореме Дейта-Фейгина находится в 4НФ
* По первой теореме Дейта-Фейгина находится в 5НФ

## Отношение Technique

Атрибуты:

* id - суррогатный ключ
* name - название техники

Функциональные зависимости:

* id -> name
* name -> id
* name -> name
* id -> id

Нормализация:

* Нет повторяющихся групп атрибутов, все атрибуты атомарны, есть ключ => находится в 1НФ
* Находится в 1НФ, нет составных ключей => находится в 2НФ
* Находится в 2НФ, все неключевые атрибуты зависят напрямую от ключей => 3НФ
* Во всех ФЗ вида X -> Y , X - надключ (рассмотрено в 1м пункте) => находится в НФБК
* По второй теореме Дейта-Фейгина находится в 4НФ
* По первой теореме Дейта-Фейгина находится в 5НФ

## Отношение TechniqueRequirement

Атрибуты:

* techniqId - внешний ключ, идентификатор техники, для которой требуется владение другой техникой (техниками)
* requiredTId - внешний ключ, идентификатор требуемой техники

Функциональные зависимости:

* Только тривиальные фз

Нормализация:

* Нет повторяющихся групп атрибутов, все атрибуты атомарны, есть ключ (составной ключ techniqId, requiredTId) => находится в 1НФ
* Нет неключевых атрибутов, для пустого множества неключевых атрибутов выполняется условие, что зависят от ключа целиком (не от части ключа) => находится в 2НФ
* Нет неключевых атрибутов, для пустого множества неключевых атрибутов выполняется условие, что зависят непосредственно от ключа => находится в 3НФ
* Нет нетривиальных ФЗ => находится в НФБК
* Отношение состоит из 2х атрибутов и очевидно не декартово произведение (для техники A не может требоваться техника A, т.е. запрещены записи вида (id, id) ) => по теореме о декомпозиции отношение находится в 4НФ
* Есть всего 2 тривиальных МЗ: {techniqId} ->> {requiredTId} | ∅, {requiredTId} ->> {techniqId} | ∅ и по теореме Фейгина можно получить 2 тривиальных СЗ => для пустого множества нетривиальных СЗ выполняется условия для 5НФ => отношение находится в 5НФ

## Отношение Skill

Атрибуты:

* ninjaId - внешний ключ, идентификатор ниндзя, которй владеет техникой с идентификатором techniqId
* techniqId - внешний ключ, идентификатор техники, которой владеет ниндзя

Функциональные зависимости:

* Только тривиальные фз

Нормализация:

* Нет повторяющихся групп атрибутов, все атрибуты атомарны, есть ключ (составной ключ techniqId, ninjaId) => находится в 1НФ
* Нет неключевых атрибутов, для пустого множества неключевых атрибутов выполняется условие, что зависят от ключа целиком (не от части ключа) => находится в 2НФ
* Нет неключевых атрибутов, для пустого множества неключевых атрибутов выполняется условие, что зависят непосредственно от ключа => находится в 3НФ
* Нет нетривиальных фз => находится в НФБК
* Отношение состоит из 2х атрибутов и очевидно не декартово произведение (как и в реальном мире с обычными людьми, все ниндзя не могут абсолютно одинаково владеть техниками) => по теореме о декомпозиции отношение находится в 4НФ
* Есть всего 2 тривиальных МЗ: {techniqId} ->> {ninjaTId} | ∅, {ninjaTId} ->> {techniqId} | ∅ и по теореме Фейгина можно получить 2 тривиальных СЗ => для пустого множества нетривиальных СЗ выполняется условия для 5НФ => отношение находится в 5НФ

## Отношение Clan

Атрибуты:

* id - суррогатный ключ
* name - уникальное название клана
* description - описание клана
* ninjaId - внешний ключ, “свидетель”, идентификатор ниндзя, будем записывать сюда основателя клана. Ниндзя может основать лишь один клан.

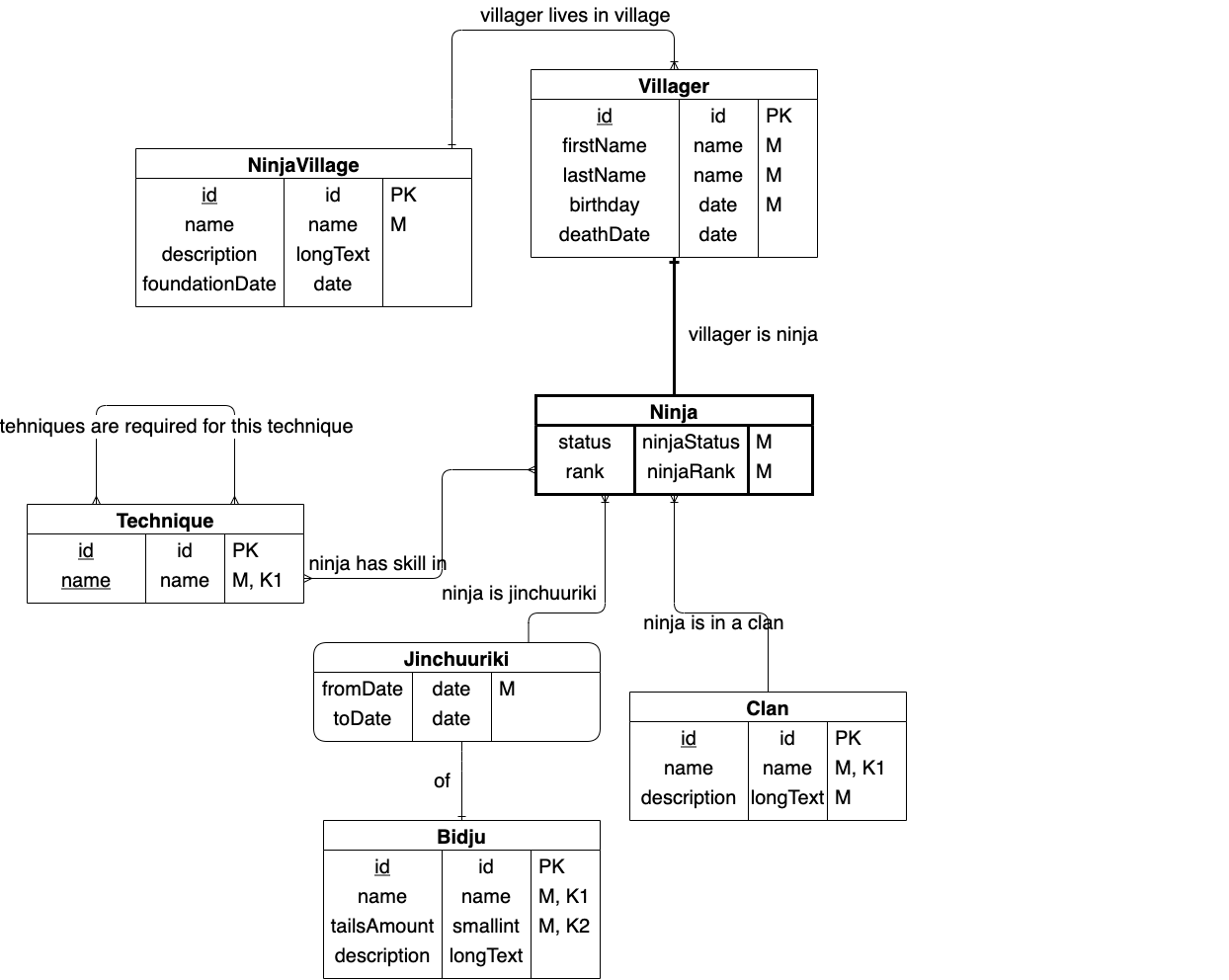
Функциональные зависимости:

* id -> name
* id -> description
* id -> ninjaId
* name -> description
* name -> id
* name -> ninjaId
* ninjaId -> name
* ninjaId -> id
* ninjaId -> description

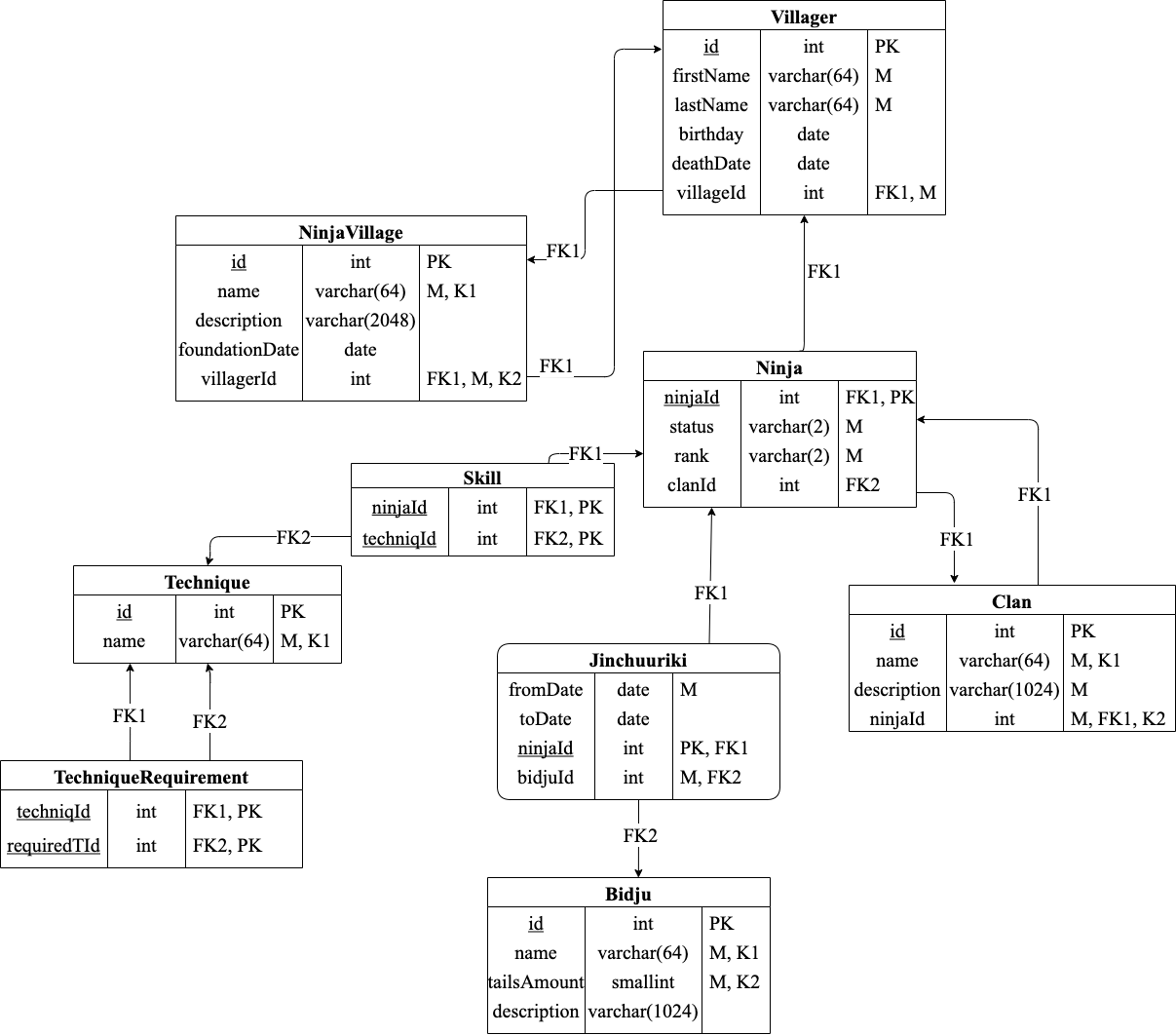
Нормализация:

* Нет повторяющихся групп атрибутов, все атрибуты атомарны, есть ключ (3 ключа: id, name, ninjaId, так как замыкание атрибутов каждого из этих атрибутов совпадает с множеством всех атрибутов) => находится в 1НФ
* Находится в 1НФ, нет составных ключей => находится в 2НФ
* Находится в 2НФ, все неключевые атрибуты зависят напрямую от ключей => находится в 3НФ
* Во всех ФЗ вида X -> Y , X - надключ (рассмотрено в 1м пункте), и во всех нетривиальных зависимостях полученных по правилам вывода левая часть тоже надключ (см. Замечание 1) => находится в НФБК
* По второй теореме Дейта-Фейгина находится в 4НФ
* По первой теореме Дейта-Фейгина находится в 5НФ

# Модель сущность-связь

Жирным выделена слабая сущность (ninja) 

# Физическая модель



При построении физической модели использовалось следующее отображение доменов в типы:

|  |  |
| --- | --- |
| Домен | Тип |
| id | int |
| name | varchar(64) |
| date | date |
| teamRank | varchar(2) |
| bool | boolean |
| longText | text(1024) |
| missionRank | varchar(2) |
| ninjaStatus | varchar(2) |
| ninjaRank | varchar(2) |
| smallint | smallint |

# Определения таблиц

Для реализации проекта использовалась СУБД postgresql 11.6 Определения таблиц и их индексов приведено в файле ddl.sql.

# Тестовые данные

Скрипт для добавления тестовых данных приведен в файле data.sql.

# Запросы на получение данных

В рамках проекта были реализованы следующие запросы:

* все жители конохи текущие - список проживающих на данный момент жителей деревни Коноха
* ниндзя конохи живые - список живых ниндзя проживающих в Конохе
* текущие джинчуурики деревни Коноха - список действующих джинчуурики деревни Коноха
* все члены клана Сенджу - список ниндзя клана Сенджу

Запросы на получение данных и вспомогательные представления приведены в файле selects.sql.

# Запросы на изменение данных

В рамках проекта были реализованы следующие запросы:

* Родиться и стать ниндзя - создание жителя и ниндзя
* Стать джинчуурики в день своего нулегого рождения - добавляет запись в таблицу Jinchuuriki, проходит валидацию в триггере
* Умереть - житель умирает, статус ниндзя переводится в ‘DD’ (dead), в таблице jibchuuriki выставляется дата to\_date. Биджу свободен
* Кто-то другой становится джинчуурики биджу - другой ниндзя становится джинчуурики этого биджу (проверка, что биджу не занят)
* Повышение ранга - просто изменение ранга
* Повышение ранга до несуществующего (упадет) - проверяет допустимые значения для ранга и падает, если нет такого значения
* Наруто не знает Элемент воздуха (упадет) - попытка выучить новую технику (новая запись в skill) и упать, так как требование не выполняется
* Упасть из-за циклической зависимости в техниках - попытка добавить запись в techniqueRequirement

Запросы на изменение данных, хранимые процедуры и триггеры приведены в файле updates.sql. Есть запросы, которые будут не проходить, в комментариях указано