**Федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

**«Саратовский государственный технический университет**

**имени Гагарина Ю. А.»**

Институт  прикладных информационных технологий и коммуникаций

Кафедра Прикладные информационные технологии

**КУРСОВАЯ РАБОТА**

по дисциплине Б.1.2.0.8 «Современные технологии управления данными в объектно-реляционных СУБД»

на тему «***Реализация объектно-ориентированной базы данных для видеоигры»***

Выполнил студент группы Б1-ИФСТ-31

Кузьмин Сергей Николаевич

Проверил: преподаватель

Кузьмин Алексей Константинович

Комиссия по защите:

доцент кафедры ПИТ Кузьмин А.К.

ассистент кафедры ПИТ Печкин И.О.

Курсовая работа защищена на оценку «\_\_\_\_\_\_\_\_\_\_\_\_\_\_»

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Кузьмин А. К.

(дата, подпись члена комиссии)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Печкин И. О.

(дата, подпись члена комиссии)

Саратов 2023

**Замечания**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Кузьмин А. К.

(дата, подпись члена комиссии)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Печкин И. О.

(дата, подпись члена комиссии)

**Федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

**«Саратовский государственный технический университет**

**имени Гагарина Ю.А.»**

Кафедра «Прикладные информационные технологии»

ЗАДАНИЕ

на выполнение курсовой работы

по дисциплине «Технологии управления данными

в объектно-реляционных СУБД»

студенту ИнПИТ группы Б1ИФСТ-31 Кузьмин Сергею Николаевичу

В курсовой работе необходимо:

Разработать базу данных в объектно-реляционной СУБД PostgreSQL для ***видеоигры***, для чего

1. Провести анализ выбранной предметной области,
2. Реализовать базу данных, используя возможности объектно-реляционной СУБД PostgreSQL
3. Протестировать базу данных на тестовых данных

Дата выдачи:    «5»    сентября 2023

Срок выполнения: «23»    декабря 2023

Руководитель:\_\_\_\_\_\_\_\_\_ Кузьмин А. К.

Студент:\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Кузьмин С. Н.

**Оглавление**

**Цель и задачи курсовой работы**

Цель курсовой работы - получение практических навыков управления и разработки баз данных в объектно-реляционной СУБД  PostgreSQL, в определенной предметной области.

Задачи курсовой работы:

1. Проектирование базы данных с более чем 6 таблицами, в которых должны быть ограничения: primary key, foreign key, default, identity, check, not null, unique.

2. Заполнение таблиц тестовыми данными.

3. Создание индексов для таблиц, чтобы увеличить производительность.

4. Реализация аналитических представлений, в которых должны быть следующие возможности SELECT: операторы группировки, итогов и подитогов, ранжирования и агрегирования (в том числе с выражением partition by), а также обобщённые (в том числе рекурсивные) табличные выражения.

5. Создание 3 курсоров и курсорных переменных различных типов.

6. Создание 2 функций, 3 процедур и 2 триггеров.

7. Создание сложных составных типов и использование их в качестве полей в таблицах и параметров в блоках.

8. Организация транзакционной обработки данных при различных установленных уровнях изолированности транзакций.

В результате выполнения курсовой работы, должны быть сформированы компетенции:

ОПК-2. Способен использовать современные информационные технологии и программные средства, в том числе отечественного производства, при решении задач профессиональной деятельности

ОПК-6. Способен представлять полученные результаты научно-исследовательской деятельности на высоком уровне и с учетом соблюдения авторских прав.

# Краткое описание используемой СУБД

PostgreSQL – это мощная объектно-реляционная система управления базами данных, обладающая рядом ключевых характеристик, делающих ее привлекательным выбором для широкого спектра приложений:

**Открытый исходный код:** PostgreSQL является открытым программным обеспечением, что позволяет пользователям свободно распространять, изменять и улучшать его. Эта особенность способствует активному сообществу разработчиков, постоянным обновлениям и высокой степени надежности.

**Расширенные возможности SQL:** СУБД поддерживает полный стандарт SQL и предлагает дополнительные возможности, такие как сложные типы данных, геопространственные запросы, а также возможности расширения собственных функций и операторов.

**Многопользовательская поддержка:** PostgreSQL обеспечивает эффективное управление множеством пользователей, предоставляя гибкие механизмы контроля доступа и обеспечивая изоляцию данных между параллельными сессиями.

**Транзакционная безопасность:** Система гарантирует ACID-свойства транзакций (Atomicity, Consistency, Isolation, Durability), обеспечивая надежность и целостность данных даже в случае сбоев.

**Масштабируемость и производительность:** PostgreSQL спроектирована для эффективной работы с различными объемами данных и обладает оптимизациями для повышения производительности в условиях высоких нагрузок.

Выбор PostgreSQL в качестве СУБД для проекта обеспечивает надежность, гибкость и расширенные возможности для эффективного управления данными.

# Реализация базы данных для видеоигры

Так как работа выполнялась в паре часть данных была создана Дуваловым Владиславом. Вот диаграмма распределения обязанностей:  


Рисунок 1. Диаграмма распределения обязаностей

Зеленым обозначена сфера обязанностей Владислава Дувалова, а желтым обозначена сфера обязанностей Сергея Кузьмина.

**Задание 1. Создание таблиц, связей и ограничений**

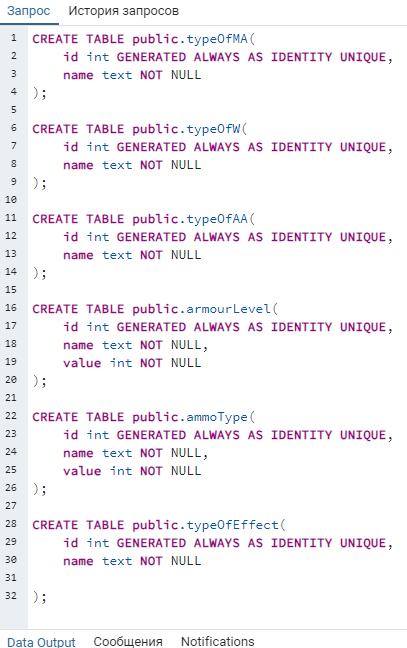


Рисунок 2 - Создание таблиц типов для других таблиц. А именно тип Способоности передвижения (MA), тип оружия (W), тип спосбности брони(AA), тип эффекта (Effect), а уровень брони (armourLevel) и тип боеприпасов (ammoType)

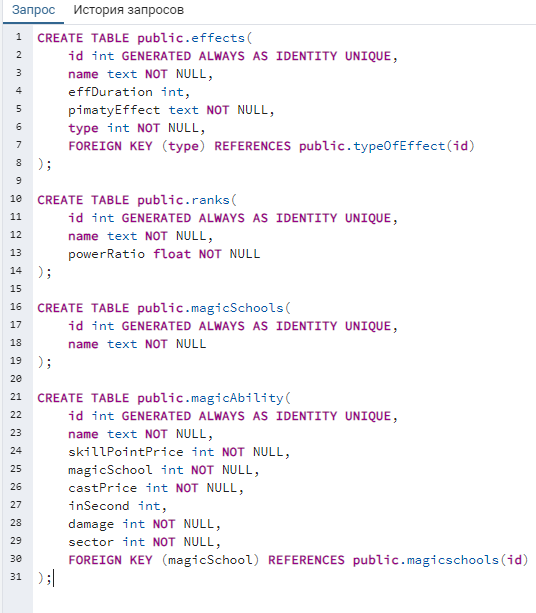


Рисунок 3 - Создание таблиц эффеков, рангов для псайкеров (магов), магических школ и заклинаний, соответственно

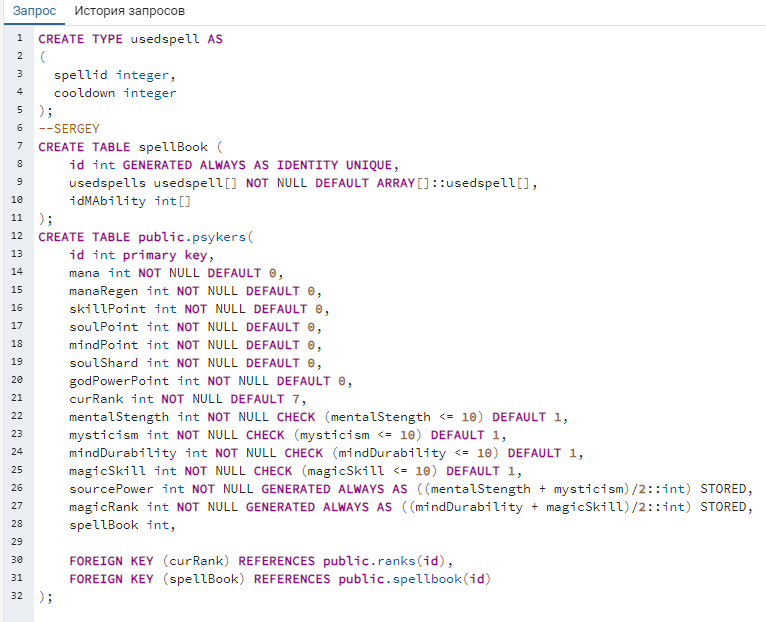


Рисунок 4 - Создание таблиц книга заклинаний и псайкера.

CREATE TABLE public.typeOfMA(

    id int GENERATED ALWAYS AS IDENTITY UNIQUE,

    name text NOT NULL

);

CREATE TABLE public.typeOfW(

    id int GENERATED ALWAYS AS IDENTITY UNIQUE,

    name text NOT NULL

);

CREATE TABLE public.typeOfAA(

    id int GENERATED ALWAYS AS IDENTITY UNIQUE,

    name text NOT NULL

);

CREATE TABLE public.armourLevel(

    id int GENERATED ALWAYS AS IDENTITY UNIQUE,

    name text NOT NULL,

    value int NOT NULL

);

CREATE TABLE public.ammoType(

    id int GENERATED ALWAYS AS IDENTITY UNIQUE,

    name text NOT NULL,

    value int NOT NULL

);

CREATE TABLE public.typeOfEffect(

    id int GENERATED ALWAYS AS IDENTITY UNIQUE,

    name text NOT NULL

);

CREATE TABLE public.effects(

    id int GENERATED ALWAYS AS IDENTITY UNIQUE,

    name text NOT NULL,

    effDuration int,

    pimatyEffect text NOT NULL,

    type int NOT NULL,

    FOREIGN KEY (type) REFERENCES public.typeOfEffect(id)

);

CREATE TABLE public.ranks(

    id int GENERATED ALWAYS AS IDENTITY UNIQUE,

    name text NOT NULL,

    powerRatio float NOT NULL

);

CREATE TABLE public.magicSchools(

    id int GENERATED ALWAYS AS IDENTITY UNIQUE,

    name text NOT NULL

);

CREATE TABLE public.magicAbility(

    id int GENERATED ALWAYS AS IDENTITY UNIQUE,

    name text NOT NULL,

    skillPointPrice int NOT NULL,

    magicSchool int NOT NULL,

    castPrice int NOT NULL,

    inSecond int,

    damage int NOT NULL,

    sector int NOT NULL,

    FOREIGN KEY (magicSchool) REFERENCES public.magicschools(id)

);

CREATE TYPE usedspell AS

(

  spellid integer,

  cooldown integer

);

--SERGEY

CREATE TABLE spellBook (

    id int GENERATED ALWAYS AS IDENTITY UNIQUE,

    usedspells usedspell[] NOT NULL DEFAULT ARRAY[]::usedspell[],

    idMAbility int[]

);

CREATE TABLE public.psykers(

    id int primary key,

    mana int NOT NULL DEFAULT 0,

    manaRegen int NOT NULL DEFAULT 0,

    skillPoint int NOT NULL DEFAULT 0,

    soulPoint int NOT NULL DEFAULT 0,

    mindPoint int NOT NULL DEFAULT 0,

    soulShard int NOT NULL DEFAULT 0,

    godPowerPoint int NOT NULL DEFAULT 0,

    curRank int NOT NULL DEFAULT 7,

    mentalStength int NOT NULL CHECK (mentalStength <= 10) DEFAULT 1,

    mysticism int NOT NULL CHECK (mysticism <= 10) DEFAULT 1,

    mindDurability int NOT NULL CHECK (mindDurability <= 10) DEFAULT 1,

    magicSkill int NOT NULL CHECK (magicSkill <= 10) DEFAULT 1,

    sourcePower int NOT NULL GENERATED ALWAYS AS ((mentalStength + mysticism)/2::int) STORED,

    magicRank int NOT NULL GENERATED ALWAYS AS ((mindDurability + magicSkill)/2::int) STORED,

    spellBook int,

    FOREIGN KEY (curRank) REFERENCES public.ranks(id),

    FOREIGN KEY (spellBook) REFERENCES public.spellbook(id)

);

Описание результата: в ходе задания 1 была построена полностью схема базы данных, а также определены зависимости, ограничения для вех таблиц. Также в ходе первого задания был создан свой тип данных, поэтому часть 7-го задания также была выполнена.

**Задание 2. Заполнение таблиц тестовыми данными**

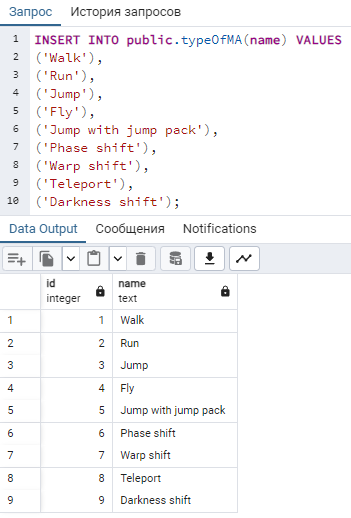


Рисунок 5 - Заполнение таблицы способов передвижения

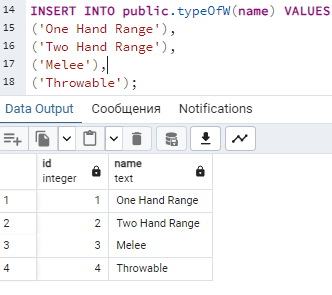


Рисунок 6 - Заполнение таблицы типов оружия

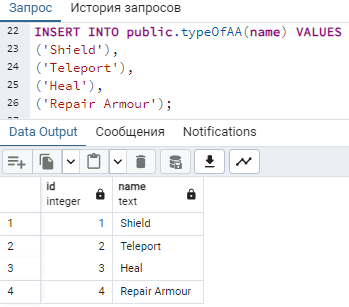


Рисунок 7 - Заполнение таблицы типов способностей брони

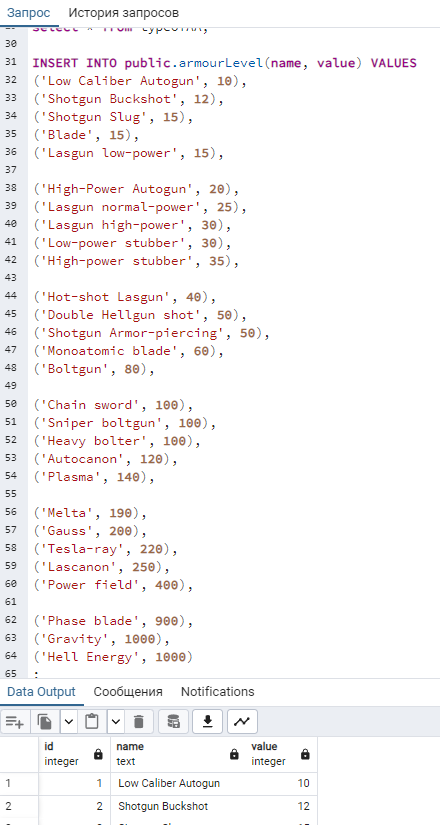


Рисунок 8 - Заполнение таблицы типов бронирования



Рисунок 9 - Заполнение таблицы типов боеприпасов

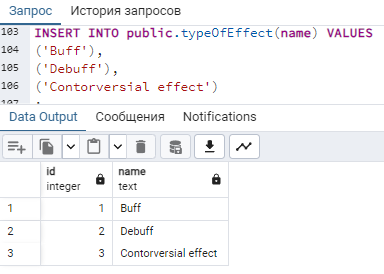


Рисунок 10 - Заполнение таблицы типов эффектов

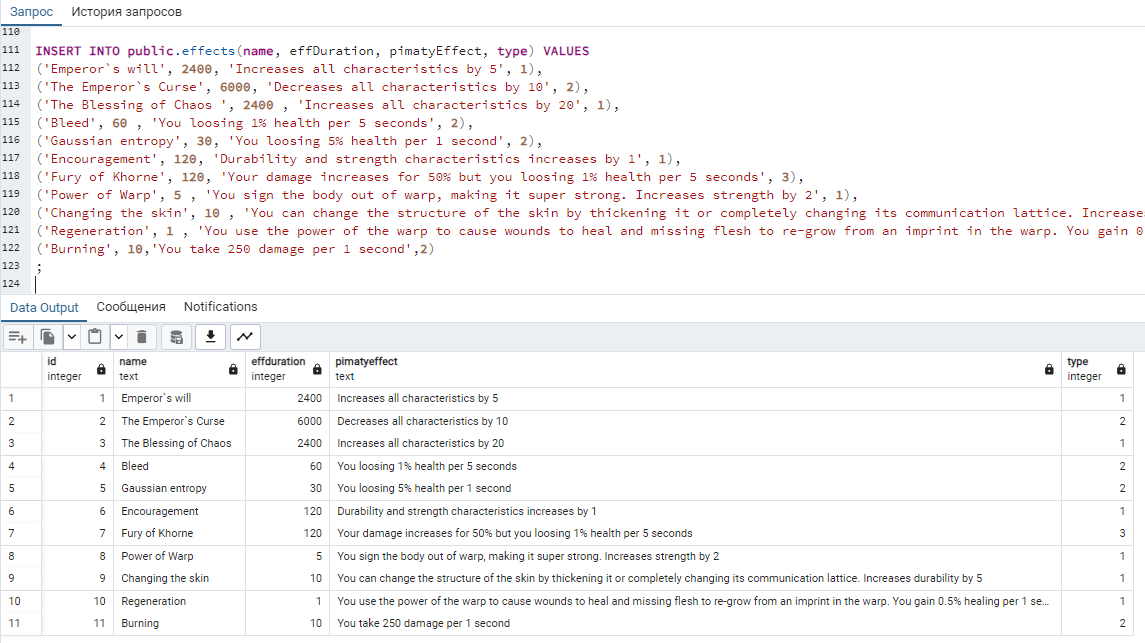


Рисунок 11 - Заполнение таблицы эффектов

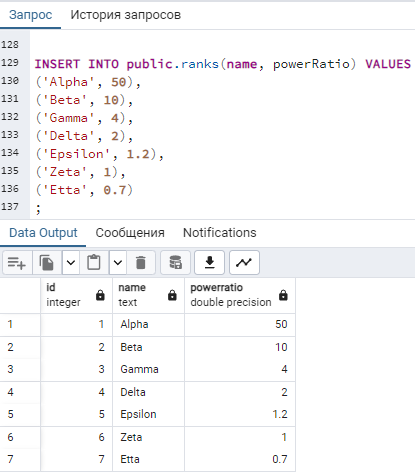


Рисунок 12 - Заполнение таблицы рангов псайкеров

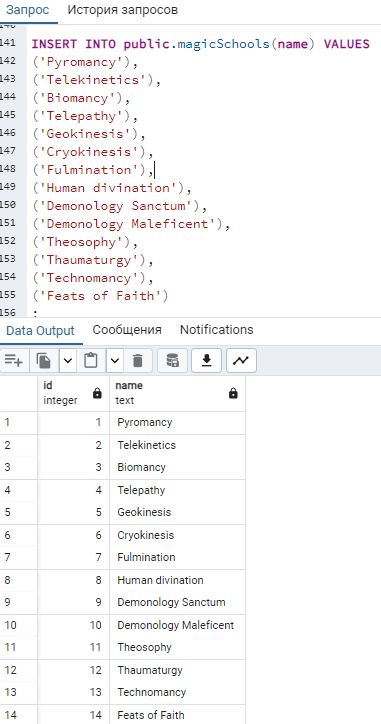


Рисунок 13 - Заполнение таблицы школ магии

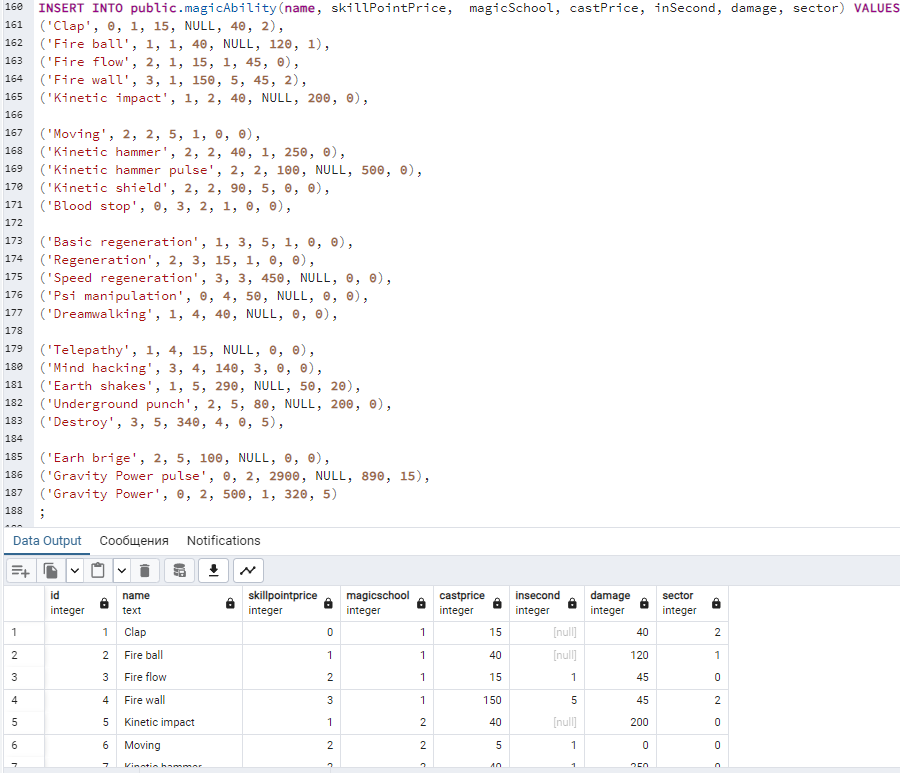


Рисунок 14 - Заполнение таблицы заклинаний

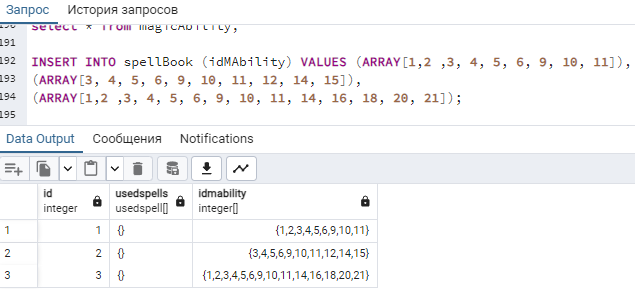


Рисунок 15 - Заполнение таблицы книг заклинаний

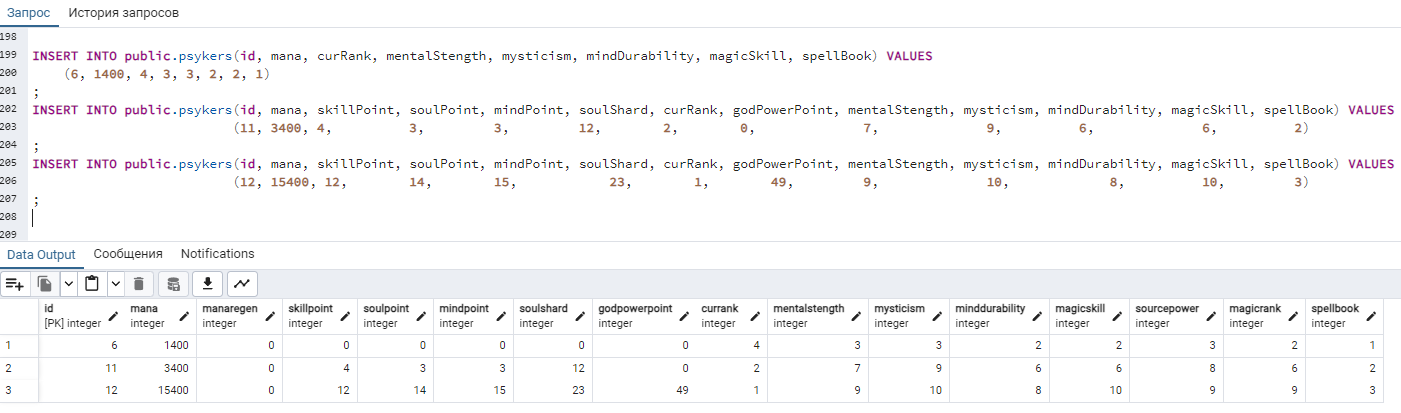


Рисунок 16 - Заполнение таблицы псайкеров

INSERT INTO public.typeOfMA(name) VALUES

('Walk'),

('Run'),

('Jump'),

('Fly'),

('Jump with jump pack'),

('Phase shift'),

('Warp shift'),

('Teleport'),

('Darkness shift');

select \* from typeOfMA;

INSERT INTO public.typeOfW(name) VALUES

('One Hand Range'),

('Two Hand Range'),

('Melee'),

('Throwable');

select \* from typeOfW;

INSERT INTO public.typeOfAA(name) VALUES

('Shield'),

('Teleport'),

('Heal'),

('Repair Armour');

select \* from typeOfAA;

INSERT INTO public.armourLevel(name, value) VALUES

('Low Caliber Autogun', 10),

('Shotgun Buckshot', 12),

('Shotgun Slug', 15),

('Blade', 15),

('Lasgun low-power', 15),

('High-Power Autogun', 20),

('Lasgun normal-power', 25),

('Lasgun high-power', 30),

('Low-power stubber', 30),

('High-power stubber', 35),

('Hot-shot Lasgun', 40),

('Double Hellgun shot', 50),

('Shotgun Armor-piercing', 50),

('Monoatomic blade', 60),

('Boltgun', 80),

('Chain sword', 100),

('Sniper boltgun', 100),

('Heavy bolter', 100),

('Autocanon', 120),

('Plasma', 140),

('Melta', 190),

('Gauss', 200),

('Tesla-ray', 220),

('Lascanon', 250),

('Power field', 400),

('Phase blade', 900),

('Gravity', 1000),

('Hell Energy', 1000)

;

select \* from armourLevel;

INSERT INTO public.ammoType(name, value) VALUES

('Low Caliber Autogun', 10),

('Shotgun Buckshot', 12),

('Shotgun Slug', 15),

('Blade', 15),

('Lasgun low-power', 15),

('High-Power Autogun', 20),

('Lasgun normal-power', 25),

('Lasgun high-power', 30),

('Low-power stubber', 30),

('High-power stubber', 35),

('Hot-shot Lasgun', 40),

('Double Hellgun shot', 50),

('Shotgun Armor-piercing', 50),

('Monoatomic blade', 60),

('Boltgun', 80),

('Chain sword', 100),

('Sniper boltgun', 100),

('Heavy bolter', 100),

('Autocanon', 120),

('Plasma', 140),

('Melta', 190),

('Gauss', 200),

('Tesla-ray', 220),

('Lascanon', 250),

('Power field', 400),

('Phase blade', 900),

('Gravity', 1000),

('Hell Energy', 1000)

;

select \* from ammoType;

INSERT INTO public.typeOfEffect(name) VALUES

('Buff'),

('Debuff'),

('Contorversial effect')

;

select \* from typeOfEffect;

INSERT INTO public.effects(name, effDuration, pimatyEffect, type) VALUES

('Emperor`s will', 2400, 'Increases all characteristics by 5', 1),

('The Emperor`s Curse', 6000, 'Decreases all characteristics by 10', 2),

('The Blessing of Chaos ', 2400 , 'Increases all characteristics by 20', 1),

('Bleed', 60 , 'You loosing 1% health per 5 seconds', 2),

('Gaussian entropy', 30, 'You loosing 5% health per 1 second', 2),

('Encouragement', 120, 'Durability and strength characteristics increases by 1', 1),

('Fury of Khorne', 120, 'Your damage increases for 50% but you loosing 1% health per 5 seconds', 3),

('Power of Warp', 5 , 'You sign the body out of warp, making it super strong. Increases strength by 2', 1),

('Changing the skin', 10 , 'You can change the structure of the skin by thickening it or completely changing its communication lattice. Increases durability by 5', 1),

('Regeneration', 1 , 'You use the power of the warp to cause wounds to heal and missing flesh to re-grow from an imprint in the warp. You gain 0.5% healing per 1 second', 1),

('Burning', 10,'You take 250 damage per 1 second',2)

;

select \* from effects;

INSERT INTO public.ranks(name, powerRatio) VALUES

('Alpha', 50),

('Beta', 10),

('Gamma', 4),

('Delta', 2),

('Epsilon', 1.2),

('Zeta', 1),

('Etta', 0.7)

;

SELECT \* FROM ranks;

INSERT INTO public.magicSchools(name) VALUES

('Pyromancy'),

('Telekinetics'),

('Biomancy'),

('Telepathy'),

('Geokinesis'),

('Cryokinesis'),

('Fulmination'),

('Human divination'),

('Demonology Sanctum'),

('Demonology Maleficent'),

('Theosophy'),

('Thaumaturgy'),

('Technomancy'),

('Feats of Faith')

;

select \* from magicSchools;

INSERT INTO public.magicAbility(name, skillPointPrice,  magicSchool, castPrice, inSecond, damage, sector) VALUES

('Clap', 0, 1, 15, NULL, 40, 2),

('Fire ball', 1, 1, 40, NULL, 120, 1),

('Fire flow', 2, 1, 15, 1, 45, 0),

('Fire wall', 3, 1, 150, 5, 45, 2),

('Kinetic impact', 1, 2, 40, NULL, 200, 0),

('Moving', 2, 2, 5, 1, 0, 0),

('Kinetic hammer', 2, 2, 40, 1, 250, 0),

('Kinetic hammer pulse', 2, 2, 100, NULL, 500, 0),

('Kinetic shield', 2, 2, 90, 5, 0, 0),

('Blood stop', 0, 3, 2, 1, 0, 0),

('Basic regeneration', 1, 3, 5, 1, 0, 0),

('Regeneration', 2, 3, 15, 1, 0, 0),

('Speed regeneration', 3, 3, 450, NULL, 0, 0),

('Psi manipulation', 0, 4, 50, NULL, 0, 0),

('Dreamwalking', 1, 4, 40, NULL, 0, 0),

('Telepathy', 1, 4, 15, NULL, 0, 0),

('Mind hacking', 3, 4, 140, 3, 0, 0),

('Earth shakes', 1, 5, 290, NULL, 50, 20),

('Underground punch', 2, 5, 80, NULL, 200, 0),

('Destroy', 3, 5, 340, 4, 0, 5),

('Earh brige', 2, 5, 100, NULL, 0, 0),

('Gravity Power pulse', 0, 2, 2900, NULL, 890, 15),

('Gravity Power', 0, 2, 500, 1, 320, 5)

;

select \* from magicAbility;

INSERT INTO spellBook (idMAbility) VALUES (ARRAY[1,2 ,3, 4, 5, 6, 9, 10, 11]),

(ARRAY[3, 4, 5, 6, 9, 10, 11, 12, 14, 15]),

(ARRAY[1,2 ,3, 4, 5, 6, 9, 10, 11, 14, 16, 18, 20, 21]);

SELECT \* FROM public.spellBook;

INSERT INTO public.psykers(id, mana, curRank, mentalStength, mysticism, mindDurability, magicSkill, spellBook) VALUES

    (6, 1400, 4, 3, 3, 2, 2, 1)

;

INSERT INTO public.psykers(id, mana, skillPoint, soulPoint, mindPoint, soulShard, curRank, godPowerPoint, mentalStength, mysticism, mindDurability, magicSkill, spellBook) VALUES

                          (11, 3400, 4,          3,         3,         12,        2,        0,              7,              9,          6,              6,          2)

;

INSERT INTO public.psykers(id, mana, skillPoint, soulPoint, mindPoint, soulShard, curRank, godPowerPoint, mentalStength, mysticism, mindDurability, magicSkill, spellBook) VALUES

                          (12, 15400, 12,        14,        15,            23,        1,        49,         9,              10,             8,          10,         3)

;

select \* from psykers;

Описание результата: Было внесено необходимое количество записей для последующего проведения тестирования.

**Задание 3. Идексирвоание**

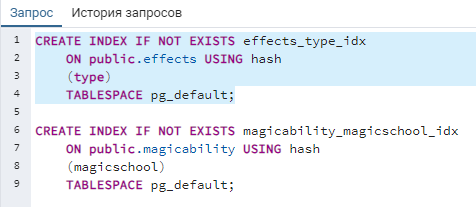
****

Рисунок 17 - Создание индеков по типу в таблице эффектов и по школе магии в заклинаниях

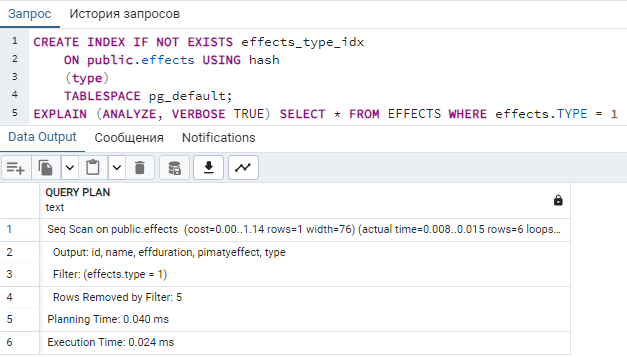


Рисунок 18 - Тестирование индекса в таблице эффектов

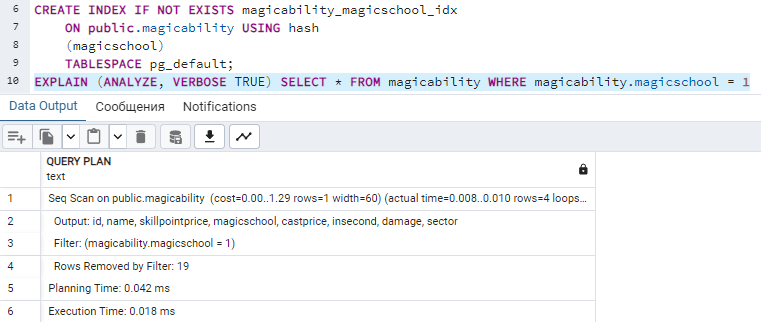


Рисунок 19 - Тестирование индекса в таблице заклинаний

CREATE INDEX IF NOT EXISTS effects\_type\_idx

    ON public.effects USING hash

    (type)

    TABLESPACE pg\_default;

EXPLAIN (ANALYZE, VERBOSE TRUE) SELECT \* FROM EFFECTS WHERE effects.TYPE = 1

CREATE INDEX IF NOT EXISTS magicability\_magicschool\_idx

    ON public.magicability USING hash

    (magicschool)

    TABLESPACE pg\_default;

EXPLAIN (ANALYZE, VERBOSE TRUE) SELECT \* FROM magicability WHERE magicability.magicschool = 1

Описание результатов: Были созданы индексы. Их эффективность доказана разницей между расчётным и фактическим временем выполнения селекта.

**Здание 4. Аналитические представления**

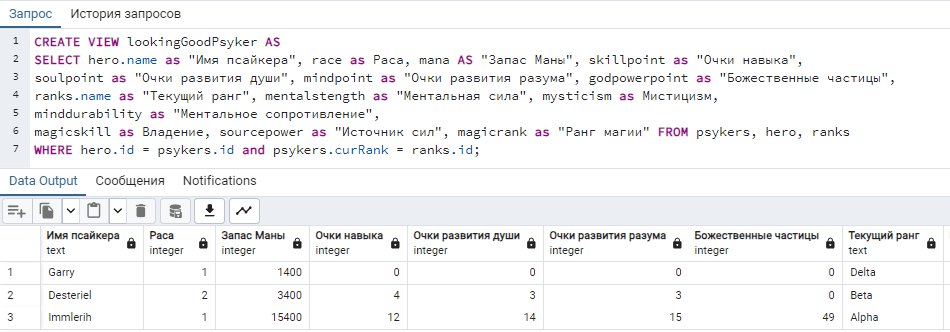
****

Рисунок 20 - Простой джоин, облегчающий восприятие данных

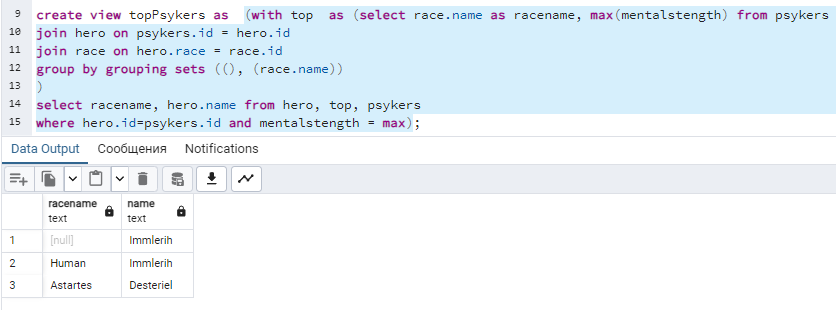


Рисунок 21 - Представления самого сильного псайкера расы

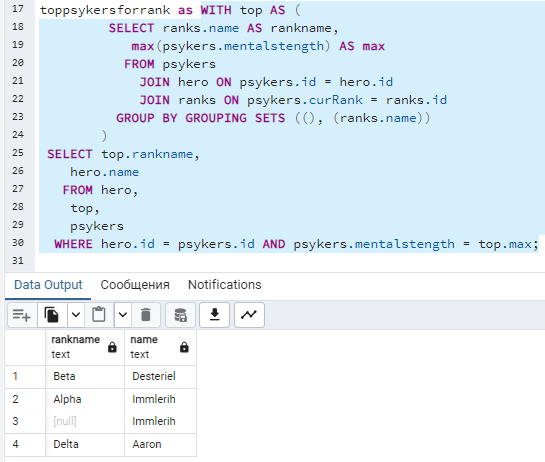


Рисунок 22 - Представление самого сильного псайкера в ранге

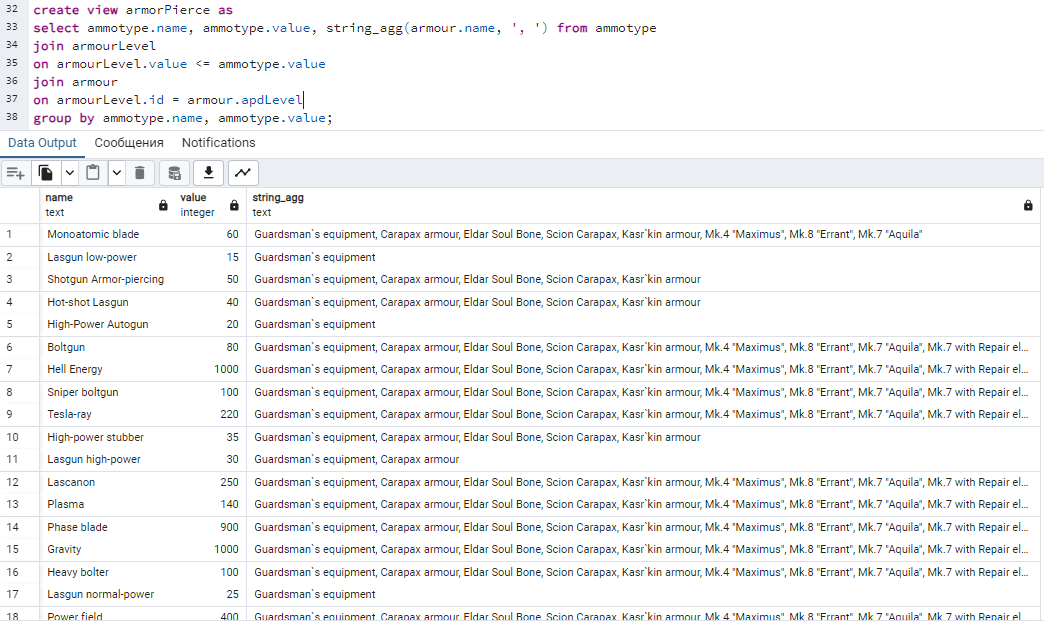


Рисунок 23 - Представление бронепробития

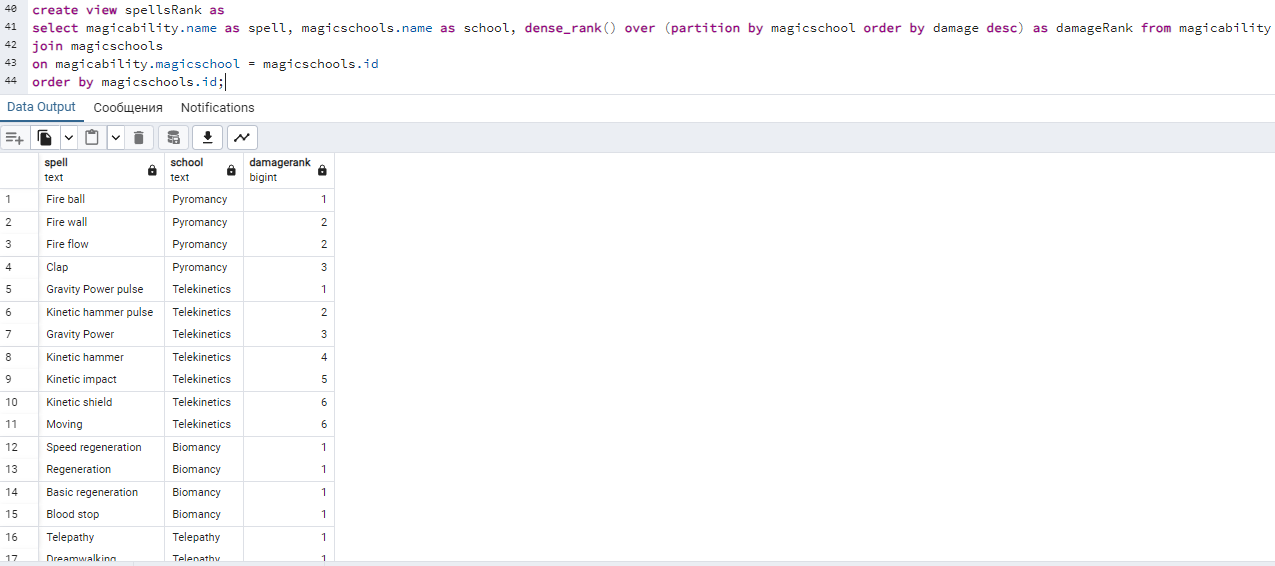


Рисунок 24 - Представление самой сильной магии в школе, сгрупированной по школе

CREATE VIEW lookingGoodPsyker AS

SELECT hero.name as "Имя псайкера", race as Раса, mana AS "Запас Маны", skillpoint as "Очки навыка",

soulpoint as "Очки развития души", mindpoint as "Очки развития разума", godpowerpoint as "Божественные частицы",

ranks.name as "Текущий ранг", mentalstength as "Ментальная сила", mysticism as Мистицизм,

minddurability as "Ментальное сопротивление",

magicskill as Владение, sourcepower as "Источник сил", magicrank as "Ранг магии" FROM psykers, hero, ranks

WHERE hero.id = psykers.id and psykers.curRank = ranks.id;

create view topPsykers as  (with top  as (select race.name as racename, max(mentalstength) from psykers

join hero on psykers.id = hero.id

join race on hero.race = race.id

group by grouping sets ((), (race.name))

)

select racename, hero.name from hero, top, psykers

where hero.id=psykers.id and mentalstength = max);

toppsykersforrank as WITH top AS (

         SELECT ranks.name AS rankname,

            max(psykers.mentalstength) AS max

           FROM psykers

             JOIN hero ON psykers.id = hero.id

             JOIN ranks ON psykers.curRank = ranks.id

          GROUP BY GROUPING SETS ((), (ranks.name))

        )

 SELECT top.rankname,

    hero.name

   FROM hero,

    top,

    psykers

  WHERE hero.id = psykers.id AND psykers.mentalstength = top.max;

create view armorPierce as

select ammotype.name, ammotype.value, string\_agg(armour.name, ', ') from ammotype

join armourLevel

on armourLevel.value <= ammotype.value

join armour

on armourLevel.id = armour.apdLevel

group by ammotype.name, ammotype.value;

create view spellsRank as

select magicability.name as spell, magicschools.name as school, dense\_rank() over (partition by magicschool order by damage desc) as damageRank from magicability

join magicschools

on magicability.magicschool = magicschools.id

order by magicschools.id;

Описание результата: Были созданы 5 представлений, каждый из которых дает ту или иную полезную информацию, отсортированную или сгруппированную тем или иным образом.

**Задание 5. Курсоры**

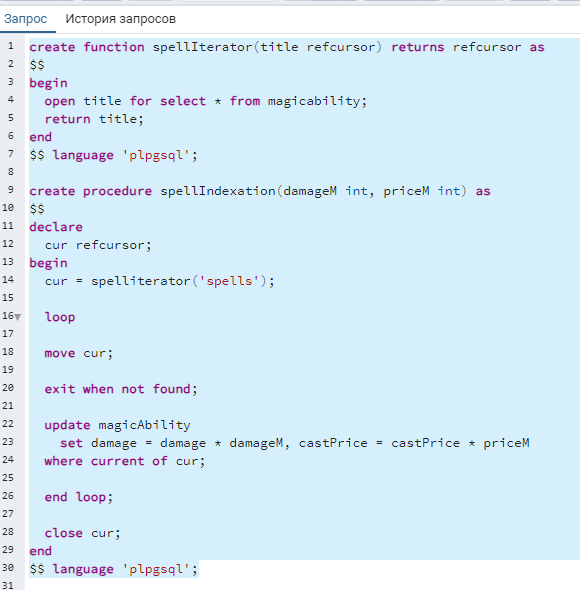
****

Рисунок 25 - Создание первой процедуры с курсором.

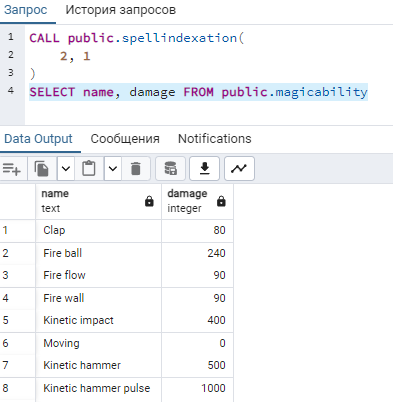


Рисунок 26 - Выполнение запроса

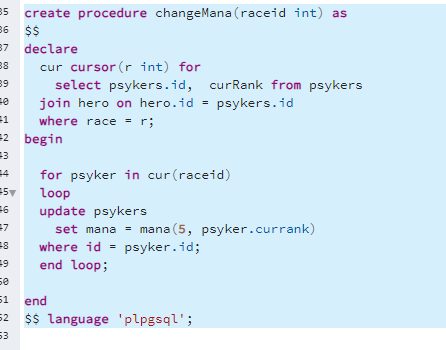


Рисунок 27 - Создание второй процедуры

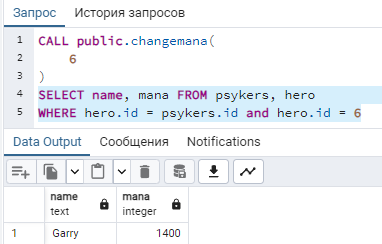


Рисунок 28 - До выполнения запроса

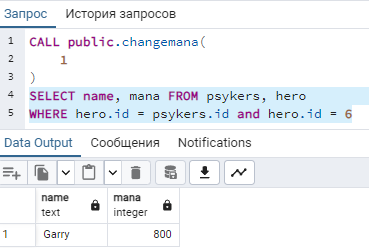


Рисунок 29 - После выполнения запроса

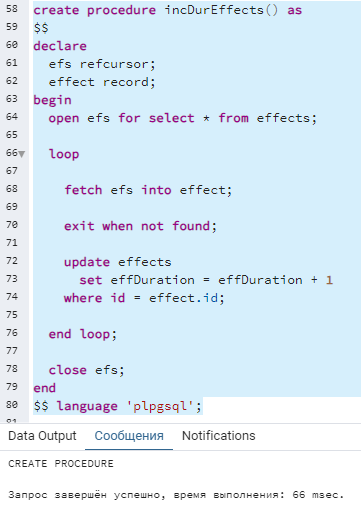


Рисунок 30 - Создание третьей процедуры

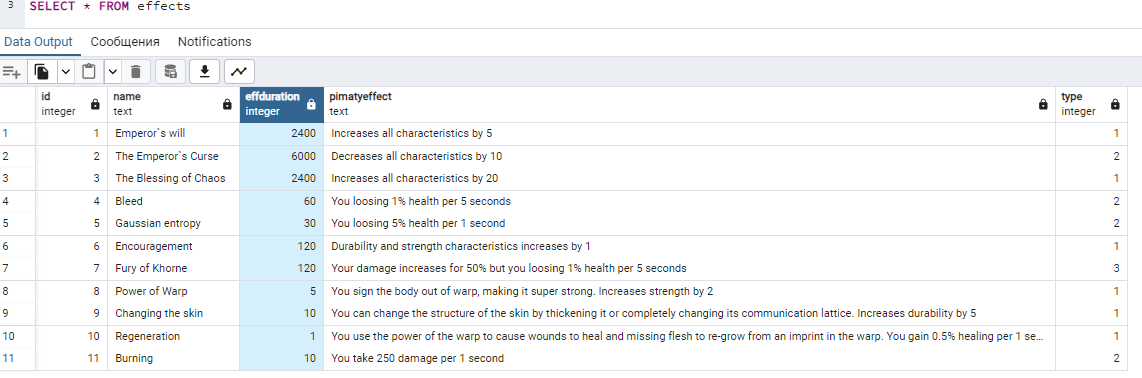


Рисунок 31- До выполнения запроса

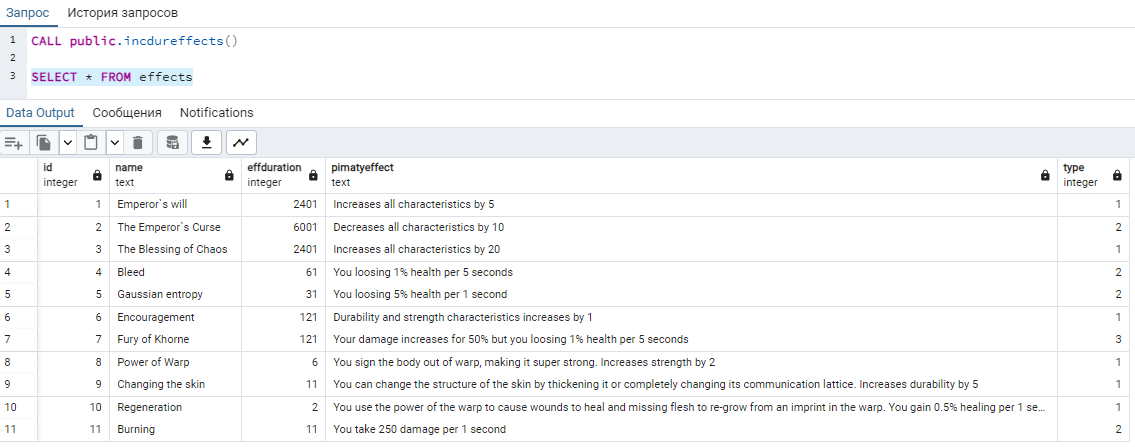


Рисунок 32 - После выполнения запроса

create function spellIterator(title refcursor) returns refcursor as

$$

begin

  open title for select \* from magicability;

  return title;

end

$$ language 'plpgsql';

create procedure spellIndexation(damageM int, priceM int) as

$$

declare

  cur refcursor;

begin

  cur = spelliterator('spells');

  loop

  move cur;

  exit when not found;

  update magicAbility

    set damage = damage \* damageM, castPrice = castPrice \* priceM

  where current of cur;

  end loop;

  close cur;

end

$$ language 'plpgsql';

create procedure changeMana(raceid int) as

$$

declare

  cur cursor(r int) for

    select psykers.id,  curRank from psykers

  join hero on hero.id = psykers.id

  where race = r;

begin

  for psyker in cur(raceid)

  loop

  update psykers

    set mana = mana(5, psyker.currank)

  where id = psyker.id;

  end loop;

end

$$ language 'plpgsql';

create procedure incDurEffects() as

$$

declare

  efs refcursor;

  effect record;

begin

  open efs for select \* from effects;

  loop

    fetch efs into effect;

    exit when not found;

    update effects

      set effDuration = effDuration + 1

    where id = effect.id;

  end loop;

  close efs;

end

$$ language 'plpgsql';

Описание результата: Были созданы 3 курсора для обновления полей таблиц и 3 процедуры в которых использовались курсоры. Курсор – это механизм обработки результирующего набора SELECT запроса.

**Задание 6. Функции, процедуры и триггеры**

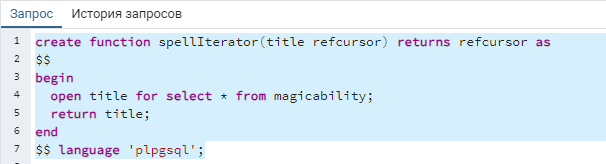


Рисунок 33 - Функция, возвращающая курсорную переменную (Использовалось ранее, см рис. 25)

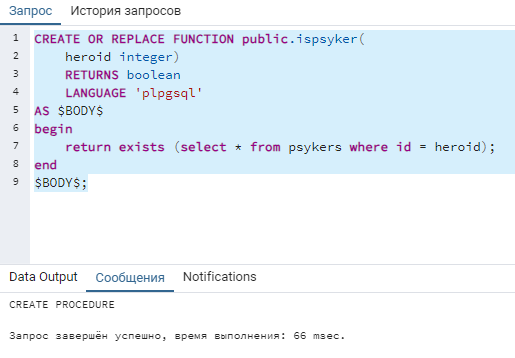


Рисунок 34 - Функция, проверяющая, псайкер ли этот персонаж

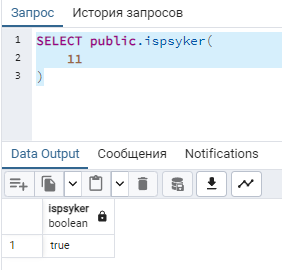


Рисунок 35 - Работа функции проверки на псайкерство

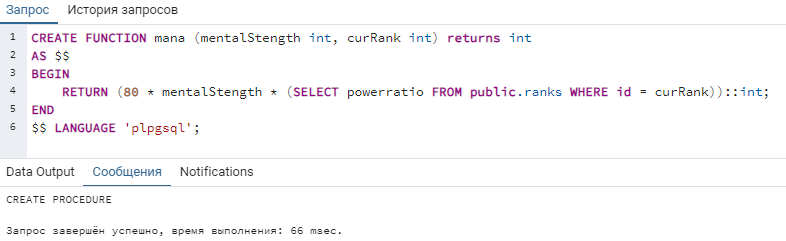


Рисунок 36 - Функция, меняющая значение параметра мана (демонстрировалось ранее, см рис 28, 29)

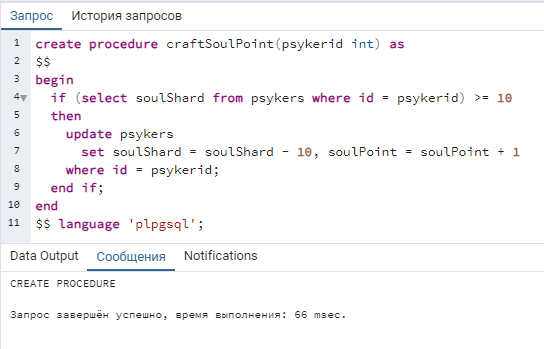


Рисунок 37 - Процедура, создающая Очко души из 10 осколков душ

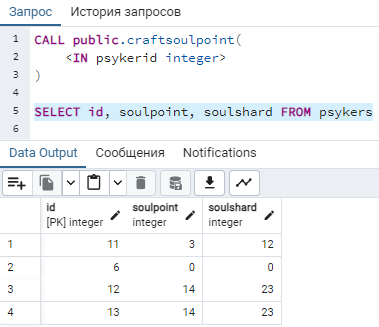


Рисунок 38 - До процедуры

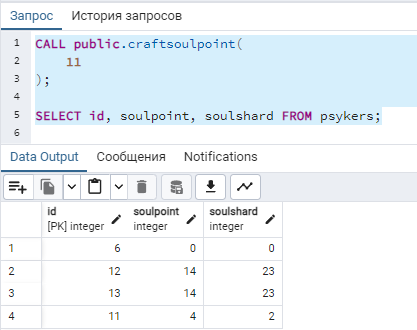


Рисунок 39 - После процедуры

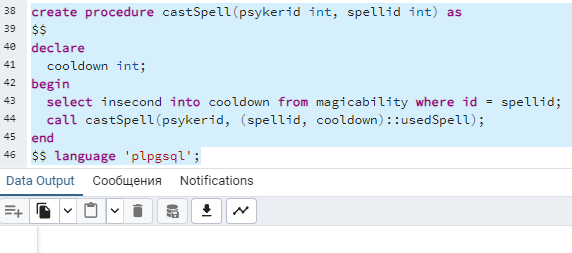


Рисунок 40 - Анти-спам процедура

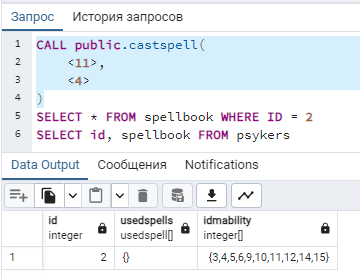


Рисунок 41- Состояние до вызова процедуры

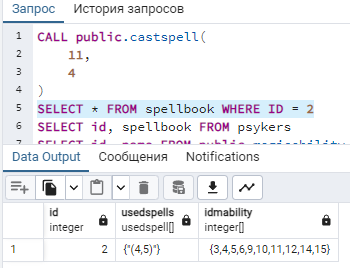


Рисунок 42 - Состояние после вызова процедуры

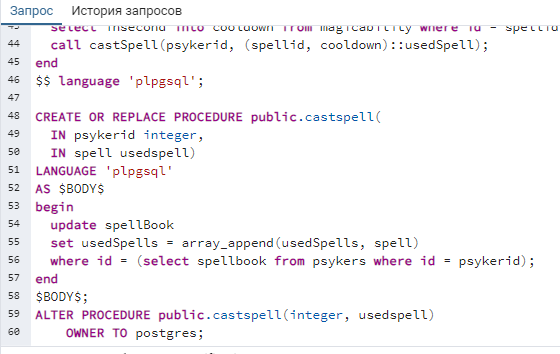


Рисунок 43 - Вспомогательная процедура

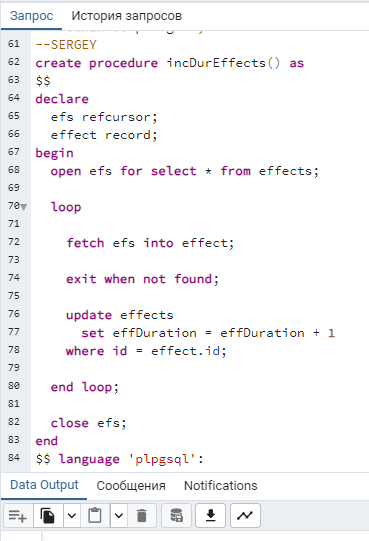


Рисунок 44 - Процедура по изменению длительности эффектов. Демонстрировалась ранее, см. рис. 30