

САНКТ – ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ФАКУЛЬТЕТ ПРИКЛАДНОЙ МАТЕМАТИКИ – ПРОЦЕССОВ УПРАВЛЕНИЯ

БАЗА ДАННЫХ “СОЗВЕЗДИЯ”

Жигалов Валентин Сергеевич

НАПРАВЛЕНИЕ 010400
ПРИКЛАДНАЯ МАТЕМАТИКА И ИНФОРМАТИКА

ПРЕПОДАВАТЕЛЬ

ФИЛИППОВ Р.О.

16 ДЕКАБРЯ 2017 Г.

Структура базы данных

Constellations:

- Id;
- Latin_name;
- genitive_latin_name;
- shortening;
- quantity_of_visible_stars;
- brightest_star;
- square;
- id_area;
- id_family.

Founders:

- id;
- name;
- country;
- date_of_birth;
- date_of_death.

Area:

- id;
- name;
- semisphere;
- max_right_ascension;
- min_right_ascension;
- max_declination;
- min_declination.

Families:

- id;
- name;
- max_right_ascension;
- min_right_ascension;
- max_declination;
- min_declination.

Constellations__founders:

- id_constellation;
- id_founder;
- id.

Areas__families:

- id_area;
- id_family;
- id.

Схема базы данных

Constellations	
Id	serial primary key
Latin_name	Varchar(30)
Genitive_latin_name	Varchar(35)
Shortening	Varchar(3)
Quantity_of_visible_stars	Integer
Brightest_star	Varchar(30)
Square	integer
Id_area	Integer
Id_family	Integer

Area	
Id	serial primary key
name	Varchar(30)
semisphere	Varchar(5)
Max_right_ascension	Varchar
Min_right_ascension	Varchar
Max_declination	Varchar
Min_declination	varchar

Areas__families	
Id_area	Integer
Id_family	integer
Id	Serial primary key

families	
Id	serial primary key
name	Varchar(30)
Max_right_ascension	real
Min_rigth_ascension	real
Max_declination	real
Min_declinaion	real

Constellations__founders	
Id_constellation	integer
Id_founder	Integer
id	Serial primary key

founders	
id	serial primary key
name	Varchar(30)
country	Varchar(15)
Date_of_birth	Integer
Date_of_death	Integer

Ссылка на репозиторий с БД: https://github.com/VZhigalov/stars_db

Описание базы данных

Данная БД представляет из себя список созвездий. В БД представлено 31 созвездие, признающееся Международным астрономическим союзом. Помимо созвездий, в таблице представлена информация об их первооткрывателях, а также семействах созвездий и областях звездного неба, в которых расположены те или иные созвездия.

Взаимоотношения между объектами

Звездное небо поделено на 8 областей (квадратов), по 4 в каждом полушарии. В каждой области расположено несколько созвездий, при этом одно созвездие не может находиться в нескольких областях звездного неба.

Семейства созвездий схожи с областями звездного неба. В них также входит несколько созвездий, при этом одно созвездие не может входить в несколько семейств. Но в отличие от областей, семейства возникали по мере открытия новых созвездий. Также созвездия из одних и тех же семейств могут быть в разных областях, и наоборот.

Что касается ученых, открывавших созвездия, то одно созвездие могло открываться несколькими учеными, если они работали вместе, или один продолжал дело другого.

Constellations:

В этой таблице представлены наиболее интересные характеристики созвездий.

Поля таблицы:

- **id primary key**

Для поля взят тип serial.

- **latin_name**

(Латинское название). Названия созвездий не могут повторяться (уникальные), и есть ограничение на размер названия (не более 30 символов). Тип varchar(30).

- **genitive_latin_name**

(Латинское название в родительном падеже). Названия созвездий в родительном падеже используются в профессиональной среде для указания, в каком созвездии расположена звезда. Также не могут повторяться и имеют ограничение на размер. Тип varchar(35).

- **shortening**

(Сокращение). Сокращения названий созвездий также используются в профессиональной среде. Больше 3 букв в сокращении быть не может. Тип varchar(3).

- **quantity_of_visible_stars**

(Количество видимых звезд). Не все звезды из созвездий можно увидеть на Земле из-за большого расстояния, а точное число всех звезд созвездия достаточно сложно высчитать. Тип integer.

- **brightest_star**

(Самая яркая звезда). Тип данных varchar(30).

- **square**

(Площадь). Площадь созвездий представлена в квадратных градусах. Квадратный угол – часть пространства, которая является объединением всех лучей, выходящих из данной точки (центра Земли) и пересекающих некоторую поверхность (звездное небо), измеряется отношением площади части сферы с центром в точке, вырезаемой квадратным углом, к квадрату радиуса сферы. Тип integer.

- **id_area**

(id области). Показывает id области, в которой расположено созвездие. Тип integer.

- **id_family**

(id семейства). Показывает id семейства, в котором расположено созвездие. Тип integer.

Founders:

Здесь представлена информация о первооткрывателях созвездий.

- **id primary key**

Тип serial.

- **name**

(Имя). Имена ученых не могут повторяться. Тип varchar(30).

- **country**

(Страна). Страна, в которой ученый жил и работал. Тип varchar(15).

- **date_of_birth**

(Дата рождения). Год рождения ученого. Взят тип integer, так как у некоторых ученых неизвестна точная дата рождения.

- **date_of_death**

(Дата смерти). Год смерти ученого. Взят тип integer, так как у некоторых ученых неизвестна точная дата смерти.

Area:

Здесь представлена информация об областях звездного неба.

- **id primary key**

Взят тип данных serial.

- **name**

(Название). Тип данных varchar(30).

- **semisphere**

(Полушарие). Поле показывает, в каком полушарии расположена данная область. Тип данных varchar(5).

- **max_right_ascension, min_right_ascension**

(Максимальное и минимальное прямое восхождение). Прямое восхождение – координата второй экваториальной системы координат, астрономический эквивалент земной долготы, но

отсчитывается от точки весеннего равноденствия. Данные в таблице являются приблизительными. Измеряется в часах, принимает значения от 0 до 24. Тип данных real.

- **max_declination, min_declination**

(Максимальное и минимальное склонение). Склонение – координата второй экваториальной системы координат, астрономический эквивалент земной широты. Данные в таблице являются приблизительными. Измеряется в градусах, принимает значения от -90 до 90. Тип данных real.

Families:

Здесь представлена информация о семействах созвездий.

- **id primary key**

Тип данных serial.

- **name**

(Название). Тип данных varchar(30).

- **max_right_ascension, min_right_ascension**

Аналогично областям.

- **max_declination, min_declination**

Аналогично областям.

Constellations__founders:

Таблица создана для реализации отношения «многие ко многим» между таблицами constellations и founders.

- **Id_constellation** – id созвездий, тип integer.
- **Id_founder** – id ученых, тип integer.
- **Id primary key** – тип serial.

Areas__families:

Таблица создана для реализации отношения «многие ко многим» между таблицами area и families.

- **id_area** – id области, тип integer.
- **id_family** – id семейства, тип integer.
- **id primary key** – тип serial.

Легкие запросы

- 1) Вывести информацию о созвездиях, отсортировав их по площади
`SELECT * FROM constellations WHERE square > 600 ORDER BY square;`
Оптимизация: `CREATE INDEX ON constellations(square);`
Создан индекс constellations_square_idx для фильтрации созвездий по площади.
- 2) Вывести имена ученых родом из Дании
`SELECT name FROM founders WHERE country = 'Denmark';`
Оптимизация: `CREATE INDEX ON founders(country text_pattern_ops);`
Создан индекс founders_country_idx для фильтрации ученых по стране.
- 3) Вывести количество видимых звезд из области с id = 1

```
SELECT sum(quantity_of_visible_stars) AS quantity_of_visible_stars
FROM constellations WHERE id_area = 1;
```

Оптимизация: `CREATE INDEX ON constellations(id_area);`

Создан индекс `constellations_id_area_idx` для фильтрации созвездий по области.

- 4) Вывести названия семейств и диапазоны их прямых восхождений и склонений
- ```
SELECT name, (max_right_ascension – min_right_ascension) AS right_ascension_diam,
(max_declination – min_declination) AS declination_diam FROM families
ORDER BY right_ascension_diam;
```

Оптимизация: не создается никаких индексов

## **Средние запросы**

- 1) Вывести информацию о созвездиях из южного полушария и отсортировать по количеству видимых звезд в созвездиях

```
SELECT latin_name, quantity_of_visible_stars, brightest_star, square, name AS area_name
FROM constellations c JOIN area a ON ((semisphere = 'South') AND (c.id_area = a.id))
ORDER BY quantity_of_visible_stars DESC;
```

Оптимизация: `CREATE INDEX ON constellations(quantity_of_visible_stars);`  
`CREATE INDEX ON area (semisphere text_pattern_ops);`  
`CREATE INDEX ON constellations(id_area);`

Создан индекс `constellations_quantity_of_visible_stars_idx` для сортировки по количеству видимых звезд.

Созданы индексы `area_semisphere_idx`, `constellations_id_area_idx`, используется индекс `area_pkey` для оптимизации объединения таблиц.

- 2) Вывести названия семейств и их площади и отсортировать по площади

```
SELECT f.name AS family, sum(c.square) AS family_square
FROM families f JOIN constellations c ON (f.id = c.id_family)
GROUP BY f.id ORDER BY family_square DESC;
```

Оптимизация: `CREATE INDEX ON area(name);`

Создан индекс `constellations_id_family_idx`, используется индекс `families_pkey` для оптимизации объединения таблиц.

- 3) Вывести id, названия семейств созвездий и количество созвездий в них и отсортировать по количеству созвездий

```
SELECT f.id, f.name AS family, count(*) AS amount_of_constellations_in_families
FROM families f JOIN constellations c ON (f.id = c.id_family)
GROUP BY f.id ORDER BY amount_of_constellations_in_families DESC;
```

Оптимизация: `CREATE INDEX ON areas__families(id_area);`

Создан индекс `constellations_id_family_idx`, используется индекс `families_pkey` для оптимизации объединения таблиц.

## **Сложные запросы**

- 1) Вывести области с количеством созвездий больше 2 и среднее арифметическое площадей созвездий области

```
SELECT name, avg_square FROM
```

```
(SELECT a.name, count(a.id) AS quantity, avg(c.square) AS avg_square
```

```
FROM constellations AS c
```

```
JOIN area AS a
```

```
ON (a.id = c.id_area)
GROUP BY a.id
ORDER BY avg_square) AS areas
WHERE quantity > 2;
```

- 2) Вывести ученого, открывшего больше всего созвездий

```
SELECT name FROM
(SELECT f.name, count(f.id) AS quantity
FROM founders AS f
JOIN constellations__founders AS c_f
ON (f.id = c_f.id_founder)
GROUP BY f.id) AS counts
WHERE quantity = (SELECT max(quantity) FROM
(SELECT f.name, count(f.id) AS quantity
FROM founders AS f
JOIN constellations__founders AS c_f
ON (f.id = c_f.id_founder)
GROUP BY f.id) AS counts);
```

- 3) Вывести названия областей с неминимальным количеством созвездий, в которых количество звезд больше 50

```
SELECT name, quantity FROM
(SELECT a.name, count(a.id) AS quantity
FROM area AS a
JOIN constellations AS c
ON (a.id = c.id_area AND c.quantity_of_visible_stars > 50)
GROUP BY a.id) AS counts
WHERE quantity != (SELECT min(quantity) FROM
(SELECT a.name, count(a.id) AS quantity
FROM area AS a
JOIN constellations AS c
ON (a.id = c.id_area AND c.quantity_of_visible_stars > 50)
GROUP BY a.id) AS counts);
```