

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО
ОБРАЗОВАНИЯ

Национальный исследовательский ядерный университет «МИФИ»



**Институт
интеллектуальных кибернетических систем**

Кафедра кибернетики (№ 22)

Направление подготовки 09.03.04 Программная инженерия

**Курсовая работа по предмету
“Основы автоматизированных информационных технологий”**

Тема: Онлайн-энциклопедия

Преподаватель: Тихомирова Дарья Валерьевна

Студент: Навроцкая Анастасия Владимировна

Группа: Б22-554

Москва 2024

1. Описание предметной области.....	3
1.1. Формулировка задания.....	3
1.2. Конкретизация предметной области.....	3
1.3. Пользователи системы.....	3
1.4. Сроки хранения информации.....	4
1.5. События, изменяющие состояние базы данных.....	4
1.6. Основные запросы к базе данных (на естественном языке).....	4
2. Концептуально-информационная модель предметной области.....	5
2.1. Er-диаграмма модели.....	5
2.2. Оценка мощностных характеристик сущностей и связей.....	5
3. Концептуальное проектирование.....	6
3.1. Принятые проектные соглашения.....	6
3.2. Обоснование выбора модели базы данных.....	6
3.3. Используемые в системе кодификаторы.....	6
3.4. Концептуальная модель базы данных.....	8
4. Логическое проектирование.....	9
4.1. Er-диаграмма базы данных.....	9
4.2. Схемы отношений базы данных.....	10
4.3. Схема реляционной базы данных.....	11
4.4. Схемы основных запросов на реляционной алгебре.....	11
1. Поиск статей по включению подстроки в названии или в тексте.....	11
5. Физическое проектирование.....	13
5.1. Обоснование выбора конкретной СУБД.....	13
5.2. Создание базы данных.....	13
5.3. Создание таблиц.....	13
5.4. Заполнение таблиц. ETL-процессы загрузки базы данных.....	14
5.5. Запросы в терминах SQL.....	19
5.6. Оценка размеров базы данных и каждого из файлов.....	24
6. Приложение.....	28
6.1. Отчеты.....	28
6.2. Создание таблиц.....	31

1. Описание предметной области

1.1. Формулировка задания

Спроектировать базу данных, содержащую информацию для редактирования и просмотра онлайн-энциклопедии: создания новых статей в организованной структуре, поддержку существующих статей с помощью внесения правок, комментирования статей и ведения обсуждений, а также коммуникацию между пользователями.

1.2. Конкретизация предметной области

Онлайн-энциклопедия содержит информационные статьи. Статья это страница, которая представляет собой озаглавленный связный текст. Страница может содержать файлы разных типов.

Также в онлайн-энциклопедии есть особые страницы - шаблоны, которые можно вставлять в другие страницы. Чаще всего шаблон полезен при многократном использовании одной и той же информации на разных вики-страницах. Шаблоны объединены в пространства имён.

Один из ключевых способов организации информации в онлайн-энциклопедии - категория. Страница может быть внесена больше чем в одну категорию. Категории могут содержать подкатегории.

Пользователями являются участники, работающие над онлайн-энциклопедией, которые могут создавать новые, а также вносить правки в существующие статьи, шаблоны, категории, комментировать статьи, участвовать в обсуждениях на страницах обсуждений. Пользователи могут быть объединены в проекты. В базе данных также сохраняется информация о просмотрах страниц.

1.3. Пользователи системы

Можно выделить три класса пользователей системы:

- Незарегистрированные пользователи. Их идентификация происходит по IP-адресу. Они могут читать все статьи и редактировать незаблокированные статьи.
- Зарегистрированные пользователи. Они также могут отмечать тип изменений, создавать статьи, изменять категории статей и загружать файлы.
- Администраторы. Их полномочия включают, кроме базовых возможностей, возможности блокировки и разблокировки аккаунтов, блокировки или разблокировки редактирования статей, редактирования заблокированных статей, удаления статей, просмотра и восстановления удаленных статей, раздачи запрашиваемых полномочий другим пользователям.

1.4. Сроки хранения информации

Основная информация, содержащаяся в онлайн-энциклопедии, должна храниться бессрочно из-за требований к возможности бессрочного доступа к статьям и их архивным версиям. После удаления статей имеет смысл сохранять в базе их архивные версии для возможности восстановления. Персональные данные удаляются через 30 дней после удаления аккаунта пользователя в соответствии с 152-ФЗ.

1.5. События, изменяющие состояние базы данных

Основные события, изменяющие хранимые в БД сущности:

- Создание страницы (статьи или категории).
- Внесение изменений в страницу, в том числе ее удаление (изменение статуса на “удалено”).
- Создание и удаление пользовательского аккаунта.
- Создание и удаление чата пользователей.

1.6. Основные запросы к базе данных (на естественном языке)

- Поиск статей по включению подстроки в названии или в тексте.
- Поиск категорий по включению подстроки в названии или в тексте описания.
- Обновление страниц при их редактировании.
- Поиск комментариев к статье для их отображения.
- Сбор статистики о самых просматриваемых статьях.
- Сбор статистики самых активных пользователей.

2. Концептуально-информационная модель предметной области

2.1. Er-диаграмма модели

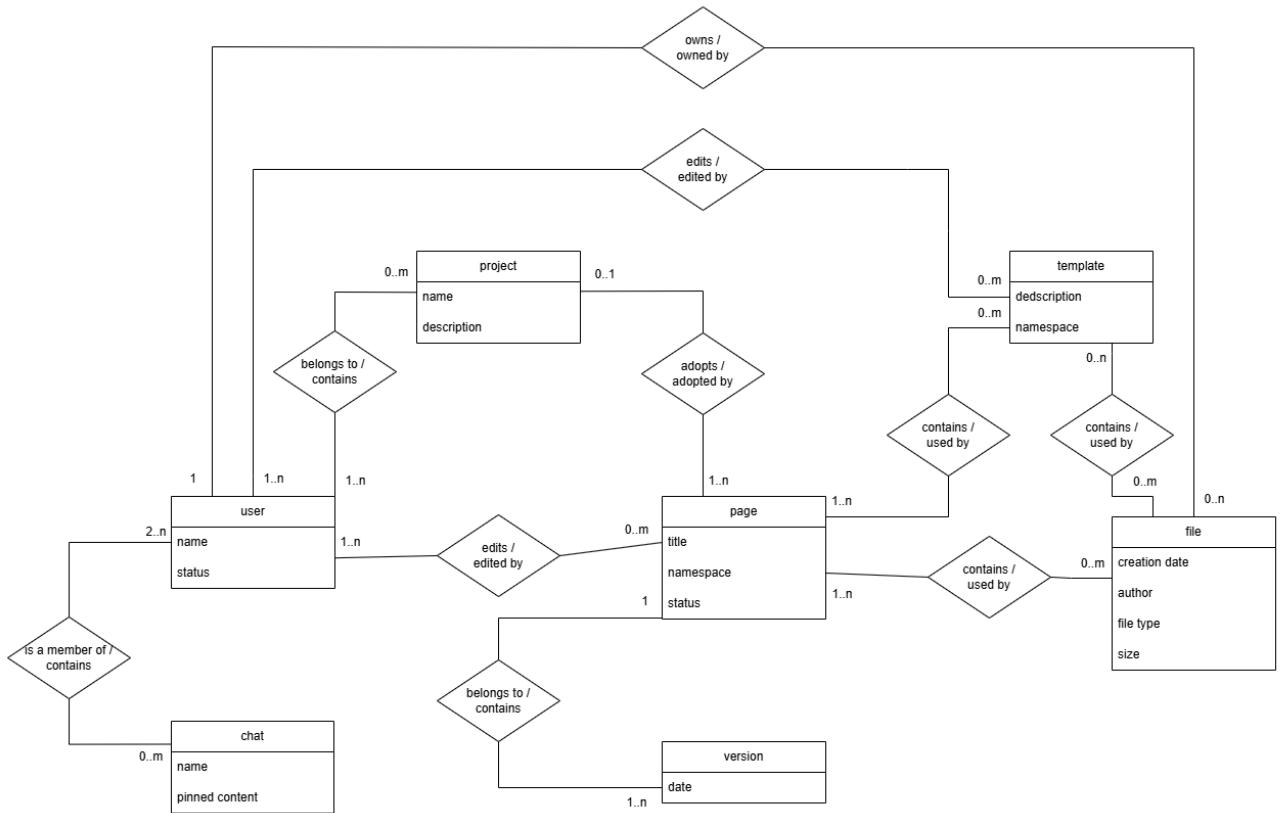


Рис. 1. ER-диаграмма модели.

2.2. Оценка мощностных характеристик сущностей и связей

Сущность / связь	Мощность		
	минимальная	средняя	максимальная
Page	1	1000	10000000
Project	0	20	1000
User	1	10000000	1000000000
Chat	0	1000	10000000
Version	1	20000	10000000
File	0	2000	1000000
Template	0	50	100000

Owns / Owned by (User - File)	0	2000	1000000
Edits / Edited by (User - Template)	0	500	1000000
Belongs to / Contains (User - Project)	0	400	100000
Adopts / Adopted by (Project - Page)	0	400	10000000
Contains / Used by (Page - Template)	0	1000	10000000
Contains / Used by (Template - File)	0	50	100000
Edits / Edited by (User - Page)	1	10000000	10000000000
Contains / Used by (Page - File)	0	2000	20000000
Is a member of / Contains (User - Chat)	0	3000	1000000000
Belongs to / Contains (Page - Version)	0	20000	100000000

3. Концептуальное проектирование

3.1. Принятые проектные соглашения

При переходе от КМПО к КМБД у каждой сущности появляется новое свойство – уникальный номер, который однозначно идентифицирует ее в базе данных.

3.2. Обоснование выбора модели базы данных

Для поставленных задач для рассматриваемой базы данных наиболее подходящей является реляционная модель, так как она позволяет извлекать из данных значимую информацию, например подсчитывать, сортировать и группировать данные в таблиц, выполнять математические операции и создавать сложные отчеты, а также является самой популярной и универсальной моделью представления данных.

3.3. Используемые в системе кодификаторы

Статус пользователя	ПО
unauthorized	Незарегистрированный пользователь
authorized	Зарегистрированный пользователь
admin	Администратор

Статус страницы	ПО
protected	Защищенная страница
stub	Болванка/заготовка страницы
deleted	Удаленная страница

Статусы пользователя и страницы хранятся в ПО, так как их значение фиксировано и не может быть изменено.

Тип файла	БД
png	Изображение
jpeg	Изображение
bmp	Изображение
txt	Текстовый файл
docx	Документ MS Word
cpp	C++ файл
zip	Архив
rar	Архив
exe	Исполняемый файл

Типы файлов хранятся в БД, так как невозможно заранее знать все возможные типы загружаемых файлов.

3.4. Концептуальная модель базы данных

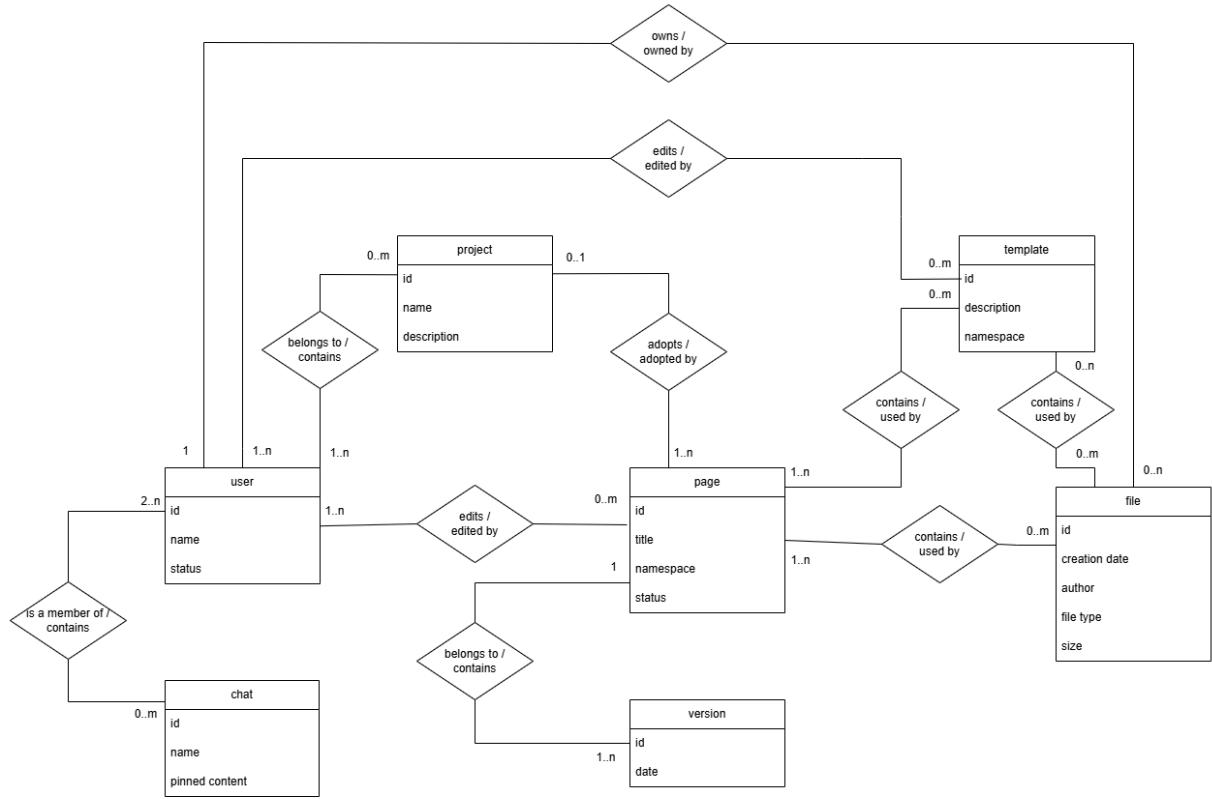


Рис. 2. Концептуальная модель базы данных.

4. Логическое проектирование

4.1. Er-диаграмма базы данных

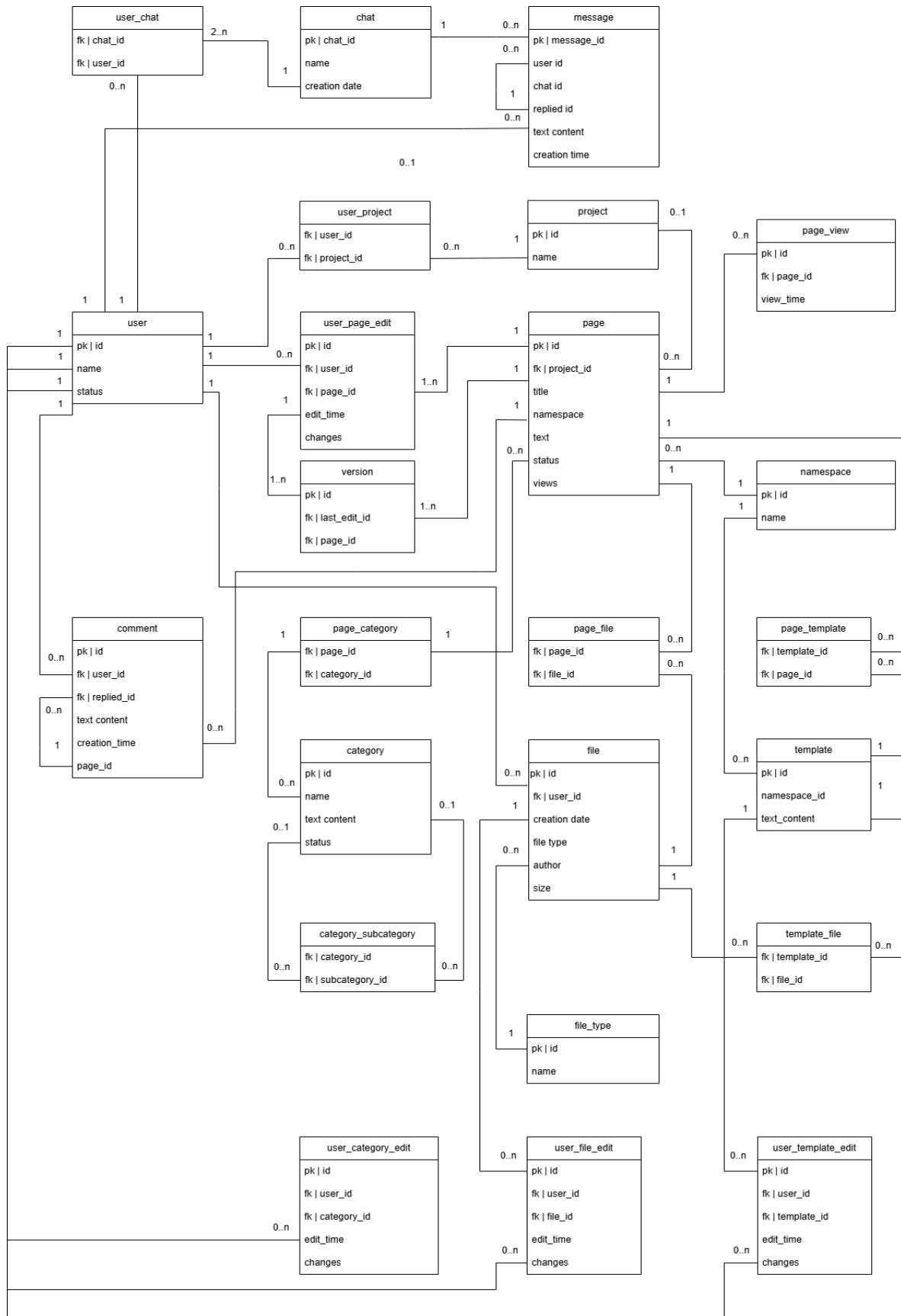


Рис. 3. ER-диаграмма базы данных.

4.2. Схемы отношений базы данных

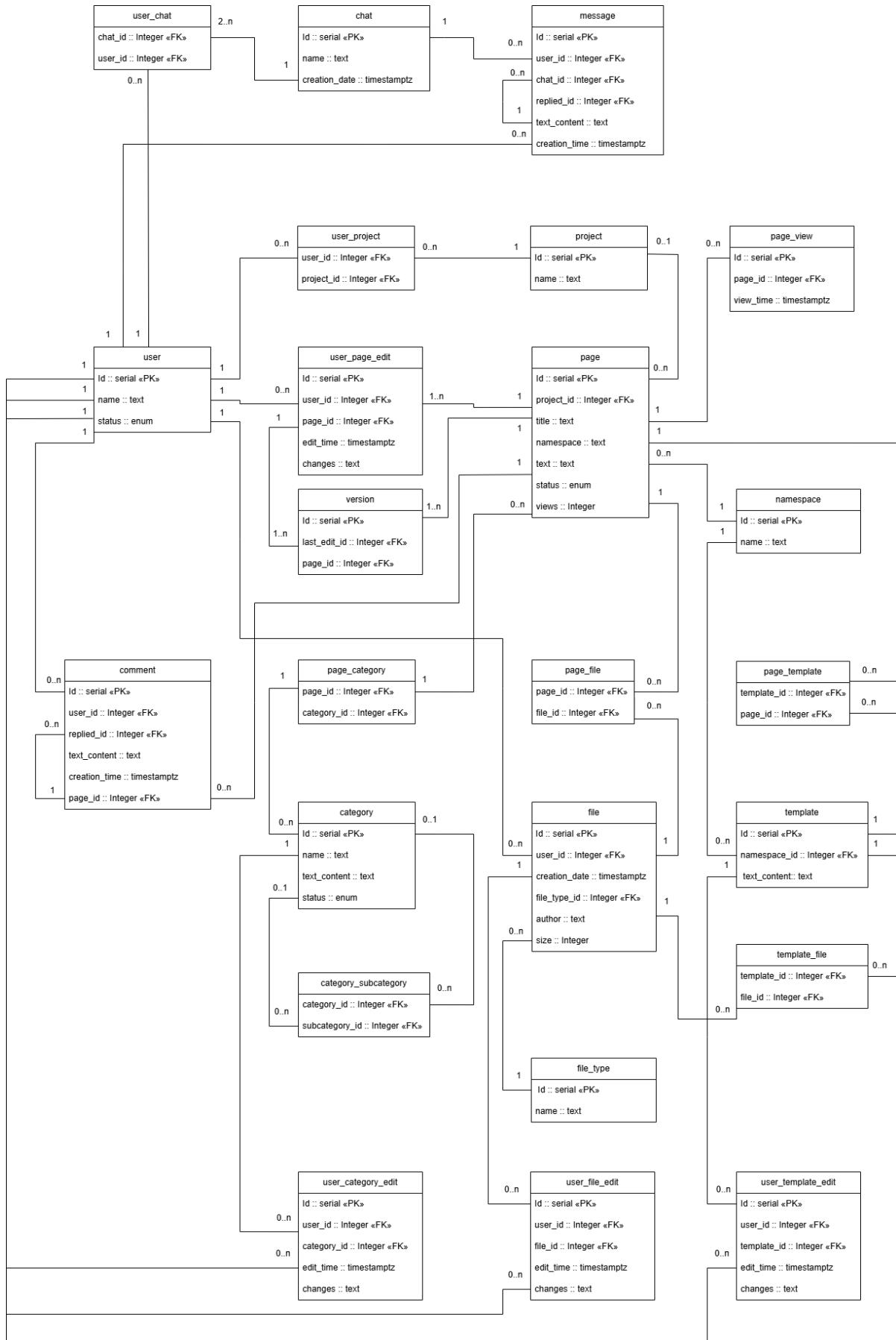


Рис. 4. Схема отношений базы данных.

4.3. Схема реляционной базы данных

R1	page (#id, project_id, title, namespace_id, status, text, views)
R2	page_category (page_id, category_id)
R3	page_template (page_id, template_id)
R4	page_file (page_id, file_id)
R5	category (#id, name, text_content, status)
R6	template (#id, namespace_id, text_content)
R7	template_file (template_id, file_id)
R8	file (#id, user_id, creation_date, file_type, author, size)
R9	category_subcategory (category_id, subcategory_id)
R10	user (#id, name, status)
R11	project (#id, name)
R12	user_page_edit (#id, user_id, page_id, changes, edit_time)
R13	user_file_edit (#id, user_id, file_id, changes, edit_time)
R14	user_category_edit (#id, user_id, category_id, changes, edit_time)
R15	user_template_edit (#id, user_id, template_id, changes, edit_time)
R16	version (#id, last_edit_id, page_id)
R17	chat (#id, name, creation_date)
R18	user_chat (#id, chat_id, user_id)
R19	comment (#id, user_id, page_id, replied_id, text_content, creation_time)
R20	message (#id, user_id, chat_id, replied_id, text_content, creation_time)
R21	namespace (#id, name)
R22	user_project (user_id, project_id)

4.4. Схемы основных запросов на реляционной алгебре

1. Поиск статей по включению подстроки в названии или в тексте.

page(id) = page[text LIKE '%example_substring%'] [id] \vee page[title LIKE '%example_substring%'] [id]

2. Поиск категорий по включению подстроки в названии или в тексте описания.

```
category(id) = category[description LIKE '%example_substring%'] [id] ∨ category[name LIKE '%example_substring%'] [id]
```

3. Обновление страниц при их редактировании.

```
page ← page - page [id == example_id] ∨ page [id, project_id, title, namespace, type, example_updated_text] [id == example_id]
```

4. Поиск комментариев к статье для их отображения.

```
comment_display(username, replied_id, text_content, creation_time) =  
((comment [page.id == example_page_id]) [comment.user_id = user.id] user)  
[user.name, comment.replied_id, comment.text_content, comment.creation_time]
```

5. Сбор статистики правок о самых просматриваемых статьях.

```
most_viewed_pages (id, title, view_count, edit_count) =  
(page ORDER BY views DESC LIMIT 100 [page.id == user_page_edit.id] user_page_edit  
GROUP BY page.id [page.id, COUNT(user_page_edit.id) as edit_count] as page_edits  
[page.edits_id == pg.id] page as pg) [page.id, page.title, page.views, page.edits.edit_count]
```

6. Сбор статистики самых активных пользователей.

```
most_active_users (id, total_edits, total_comments) =  
(user[user.id == user_page_edit.id] user_page_edit.id[user.id == comment.user_id] comment  
GROUP BY user.id) [page.id, COUNT(user_page_edit.id), COUNT(comment.id)]
```

5. Физическое проектирование

5.1. Обоснование выбора конкретной СУБД

В качестве СУБД для физической реализации спроектированной базы данных была выбрана PostgreSQL. Она хорошо подходит для целей по созданию крупномасштабных БД, требующих высокой скорости чтения и записи данных. Она очень популярна в сообществе разработчиков. PostgreSQL - бесплатная система с открытым исходным кодом. PostgreSQL может работать на многих платформах включая Linux, Windows, Mac OS.

5.2. Создание базы данных

Для создания БД использовалось клиентское программное приложение DBeaver.
Подключение к БД: connection: postgres localhost:5432, database name: postgres, schema: public.

5.3. Создание таблиц

```
CREATE TABLE public.page (
    id int8 NOT NULL GENERATED ALWAYS AS IDENTITY,
    title varchar NULL,
    project_id int8 NULL,
    "views" int8 NULL,
    status public.page_status NULL,
    namespace_id int8 NULL,
    "text" text NULL,
    CONSTRAINT pk_page_id PRIMARY KEY (id)
);
CREATE INDEX fki_fk_project_id ON public.page USING btree
(project_id);
ALTER TABLE public.page ADD CONSTRAINT fk_namespace_id FOREIGN KEY
(namespace_id) REFERENCES public."namespace"(id);
ALTER TABLE public.page ADD CONSTRAINT fk_project_id FOREIGN KEY
(project_id) REFERENCES public.project(id);
```

Листинг 1. Создание таблицы page.

Код создания других таблиц приведен в приложении. Создание проводилось в СУБД PostgreSQL.

5.4. Заполнение таблиц. ETL-процессы загрузки базы данных

Таблица Category

id	name	text_content	status
1	Bird	Birds are a group of warm-blooded vertebrates constituting the class Aves.	protected
2	Fish	A fish is an aquatic, anamniotic, gill-bearing vertebrate animal with swimming fins and a	protected
3	Cat	The cat (<i>Felis catus</i>), also referred to as domestic cat or house cat, is a small domesticated	protected
4	Fruit	In botany, a fruit is the seed-bearing structure in flowering plants (angiosperms) that is fo	protected
5	Vegetable	Vegetables are parts of plants that are consumed by humans or other animals as food.	deleted
6	Music	Music is the arrangement of sound to create some combination of form, harmony, melody, stub	
7	Country	A country is a distinct part of the world, such as a state, nation, or other political entity. W	protected
8	City	A city is a human settlement of a substantial size	protected
9	Person	A street is a public thoroughfare in a built environment.	deleted
10	Job type	A job type defines the accounting behavior for the related job.	deleted

Рис.5. Результат заполнения таблицы Category.

Таблица Category_subcategory

	category_id	subcategory_id
1	1	11
2	1	12
3	1	13
4	1	14
5	1	15
6	1	16
7	1	17
8	1	18
9	1	19
10	1	20

Рис.6. Результат заполнения таблицы Category_subcategory.

Результаты заполнения остальных таблиц приведены в приложении. Заполнение проводилось с помощью [faker.js](#).

5.5. Запросы в терминах SQL

1. Поиск статей по включению подстроки в названии или в тексте.

```
select
    p.id as page_id
from page p
where text like '%example_substring%'
    or title like '%example_substring%';
```

Листинг 2. Поиск статей по включению подстроки.

SQL SELECT id FROM page WHERE text LIKE '%cas'

	id
1	424
2	582
3	950
4	1,733
5	5,032
6	6,110
7	7,707
8	7,992
9	8,455
10	8,615
11	9,706

Рис.7. Результат выполнения запроса 1.

2. Поиск категорий по включению подстроки в названии или в тексте описания.

```
select
    c.id as category_id,
    c."name" as category_name,
    c.text_content
from category c
where text_content like '%animal%'
    or name ilike '%animal%';
```

Листинг 3. Поиск категорий по включению подстроки.

SQL select c.id as category_id, c."name" as category_name, Enter a SQL expression to filter results (use Ctrl+Space)

category_id	category_name	text_content
1	Vegetable	Vegetables are parts of plants that are consumed by humans or other animals as food.
2	Fish	A fish is an aquatic, anamniotic, gill-bearing vertebrate animal with swimming fins and a hard skull, but

Рис.8. Результат выполнения запроса 2.

3. Поиск комментариев к статье для их отображения.

```
select
    u.name AS user_name,
    c.replied_id,
    c.text_content,
    c.creation_time,
    p.id as page_id,
```

```

    p.title as page_title
from comment c
join page p on
    c.page_id = p.id
join "user" u on
    c.user_id = u.id
where p.id = 1364;

```

Листинг 4. Поиск комментариев к статье.

	user_name	replied_id	text_content	creation_time	page_id	page_title
1	Wilfrid	794	Practical Hat designed with Steel	2024-11-13 14:31:50.000	1,364	airline
2	Augusta	3,267	Our frog-friendly Computer ensures	2023-12-11 08:38:30.000	1,364	airline

Рис.9. Результат выполнения запроса 3.

4. Сбор статистики правок о самых просматриваемых статьях.

```

select
    p.id as page_id,
    p.title,
    p.views as view_count,
    count(upe.id) as edit_count
from page p
left join user_page_edit upe on
    p.id = upe.page_id
group by p.id
order by p.views desc
limit 100;

```

Листинг 5. Сбор правок самых просматриваемых статей.

SELECT p.id, p.title, p.views AS view_count, COI					Enter a SQL expression to filter results (use Ctrl+Space)
	id	title	view_count	edit_count	
1	1,196	haversack	99,971	2	
2	6,848	impostor	99,958	0	
3	3,651	innovation	99,956	1	
4	3,897	outrun	99,931	1	
5	3,937	veto	99,931	3	
6	1,560	monasticism	99,929	2	
7	9,000	orchid	99,929	2	

Рис.10. Результат выполнения запроса 4.

5. Сбор статистики самых активных пользователей в наборе проектов.

```
select
    u.id as user_id,
    u.name,
    count(distinct upe.id) as total_edits,
    count(distinct c.id) as total_comments,
    count(distinct upe.id) + count(distinct c.id) as
total_activity
from
    "user" u
left join user_page_edit upe on
    u.id = upe.user_id
left join comment c on
    u.id = c.user_id
left join user_project up on
    up.user_id = u.id
left join project p on
    p.id = up.project_id
where p."name" in ('Intelligent Wooden Gloves', 'Luxurious
Concrete Bike', 'Recycled Steel Shoes')
group by
    u.id, u.name
having count(distinct upe.id) >= 0
order by total_activity desc, u.name asc
```

Листинг 6. Сбор статистики активных пользователей в наборе проектов.

	user_id	name	total_edits	total_comments	total_activity
1	8,364	Edwardo	3	2	5
2	8,935	Jailyn	1	4	5
3	2,298	Felicia	4	0	4
4	637	Braden	1	2	3
5	9,911	Leonardo	2	1	3
6	5,060	Orion	0	3	3
7	6,558	Chad	1	1	2
8	7,551	Maybelle	2	0	2
9	9,401	Carmine	0	1	1
10	8,420	Lindsay	0	1	1
11	5,816	Bernhard	0	0	0

Рис.10. Результат выполнения запроса 5.

6. Количество добавленных статей по месяцам с дополнительными условиями.

Сколько статей добавляется в процентах по месяцам со статусом ‘protected’ для категорий со статусом ‘protected’, у которых изменения статей и категорий производились не позднее, чем 1 января 2024 года. Учитывать правки пользователей, находящихся в статусе 'administrator', 'member', 'patrolling'.

```

select

    case creation_month
        when 1 then 'January'
        when 2 then 'February'
        when 3 then 'March'
        when 4 then 'April'
        when 5 then 'May'
        when 6 then 'June'
        when 7 then 'July'
        when 8 then 'August'
        when 9 then 'September'
        when 10 then 'October'
        when 11 then 'November'
        when 12 then 'December'
    end as month,
    count(page_id) as total_per_month
from
    (select

```

```
upe.page_id,
count(u.id),
extract (month from min(upe.edit_time)) as creation_month
from user_page_edit up
join page_category pc on
    upe.page_id = pc.page_id
join category c on
    c.id = pc.category_id
join page p on
    p.id = upe.page_id
join user_category_edit uce on
    uce.category_id = c.id
join public.user u on
    u.id = uce.user_id
where c.status = 'protected'
    and p.status = 'protected'
    and upe.edit_time >= '2024-01-01'
    and uce.edit_time >= '2024-01-01'
    and u.status in ('administrator', 'member',
'patrolling')
group by upe.page_id
having count(u.id) > 10
order by creation_month) as page_creation_month
group by creation_month
```

Листинг 7. Количество добавленных статей по месяцам с дополнительными условиями.

	month	total_per_month
1	January	631
2	February	529
3	March	637
4	April	610
5	May	520
6	June	519
7	July	465
8	August	480
9	September	450
10	October	404
11	November	375
12	December	691

Рис.11. Результат выполнения запроса 6.

5.6. Оценка размеров базы данных и каждого из файлов

Отношение	Атрибут	Тип данных	Размер, байт	Среднее количество	Объем, байт
category	id	int8	8	1000	1048000
	name	text	~ 20		
	text_content	text	~ 1000		
	status	enum	4		
category_subcategory	category_id	int8	8	1000	16000
	subcategory_id	int8	8		
comment	id	int8	8	10000000	104000000000
	user_id	int8	8		
	replied_id	int8	8		
	text_content	text	~ 1000		
	creation_time	timestamp	8		
	page_id	int8	8		
file	id	int8	8	2000	112000
	user_id	int8	8		
	creation_time	timestamp	8		

	author	text	~ 20		
	size	int8	8		
	file_type_id	int4	4		
file_type	id	int8	8	100	2800
	name	text	~ 20		
namespace	id	int8	8	1000	28000
	name	text	~ 20		
page	id	int8	8	1000	40056000
	title	text	~ 20		
	project_id	int8	8		
	views	int8	8		
	status	page_status	4		
	namespace_id	int8	8		
	text	text	~ 40000		
page_category	page_id	int8	8	1000	16000
	category_id	int8	8		
page_file	page_id	int8	8	2000	32000
	file_id	int8	8		
page_template	page_id	int8	8	1000	16000
	template_id	int8	8		
page_view	it	int8	8	10000000	240000000
	page_id	int8	8		
	view_time	timestamp	8		
project	id	int8	8	20	280
	name	text	~ 20		
status	id	int8	8	10	280
	name	text	~ 20		
template	id	int8	8	50	500800
	namespace_id	int8	8		

	text_content	text	~ 10000		
template_file	template_id	int8	8	50	1450
	file_id	int8	8		
user	id	int8	8	10000000	410000000
	name	text	~ 20		
	project_id	int8	8		
	status	user_status	4		
user_category_edit	id	int8	8	2000	66000
	user_id	int8	8		
	category_id	int8	8		
	edit_time	timestamp	8		
	deleted	bool	1		
user_file_edit	id	int8	8	2000	5200
	user_id	int8	8		
	file_id	int8	8		
	deleted	bool	1		
user_page_edit	id	int8	8	10000000	330000000
	user_id	int8	8		
	page_id	int8	8		
	edit_time	timestamp	8		
	deleted	bool	1		
user_project	user_id	int8	8	400	6400
	project_id	int8	8		
user_template_edit	id	int8	8	500	16500
	user_id	int8	8		
	file_id	int8	8		
	edit_time	timestamp	8		
	deleted	bool	1		
version	id	int8	8	20000	480000

	last_edit_id	int8	8		
	page_id	int8	8		
chat	id	int8	8	1000	36000
	name	text	~ 20		
	creation_date	timestamp	8		
user_chat	user_id	int8	8	10000	160000
	chat_id	int8	8		
message	id	int8	8	10000	5400000
	user_id	int8	8		
	chat_id	int8	8		
	replied_id	int8	8		
	text_content	text	~ 500		
	creation_time	timestamp	8		

Средний размер базы данных 11421923710 байта или 10.64 ГБ.

6. Приложение

6.1. Отчеты

Графики были построены на основе Excel. В качестве отчетов, которые будет генерировать система, были взяты следующие:

1. Количество добавленных статей по месяцам с дополнительными условиями

Сколько статей добавляется в процентах по месяцам со статусом 'protected' для категорий со статусом 'protected', у которых изменения статей и категорий производились не позднее, чем 1 января 2024 года. Учитывать правки пользователей, находящихся в статусе 'administrator', 'member', 'patrolling'.

См. листинг 7.

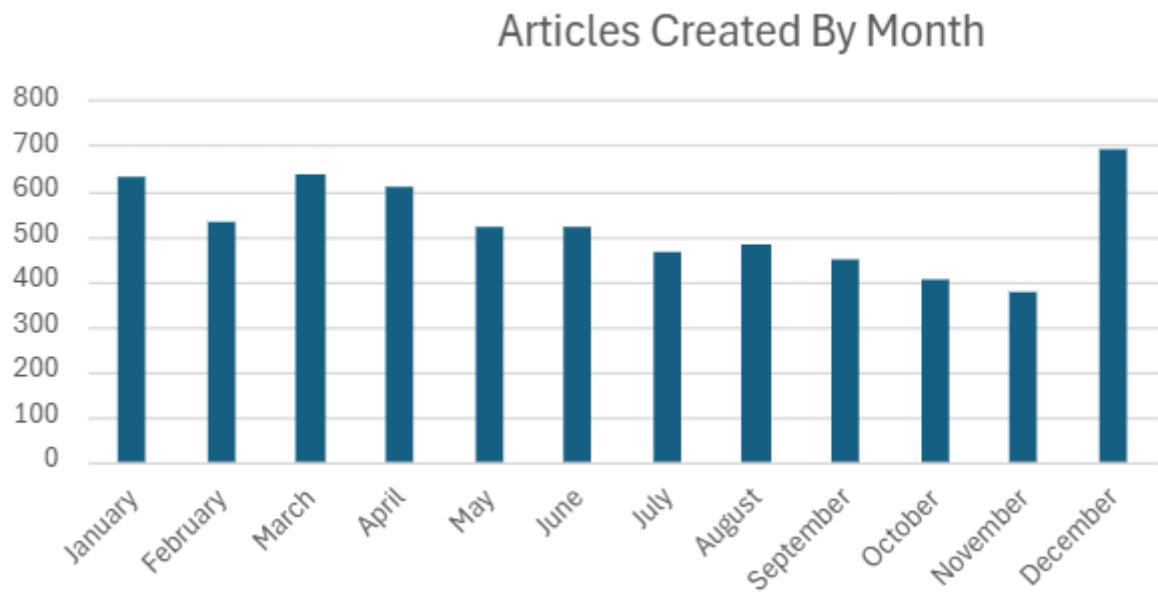


Рис.12. Результат выполнения запроса.

2. Количество правок в страницах с начала года по месяцам

```
select date_part('month', edit_time), COUNT(id) from
user_page_edit as edit_number
where (edit_time >= '2025-01-01' and edit_time <= '2025-05-21')
group by date_part('month', edit_time)
```

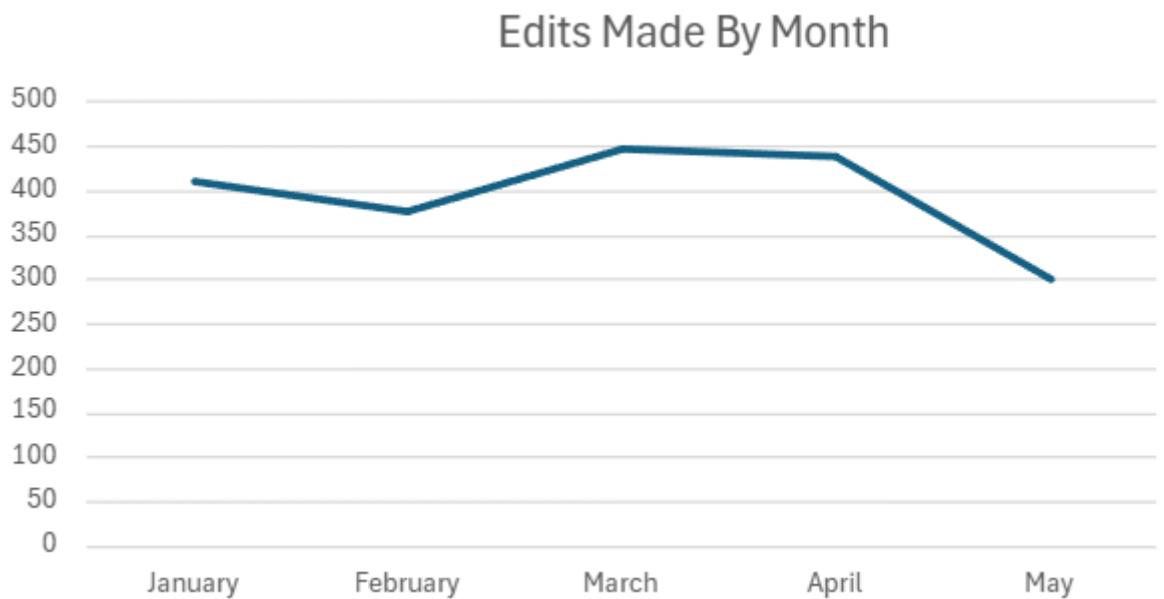


Рис.13. Результат выполнения запроса.

3. Количество правок, сделанных в целом каждой категорией пользователей

```
select
    u.status as user_status,
    count(upe.id) as user_edits
from user_page_edit upe
join "user" u on
    u.id = upe.user_id
group by status
```

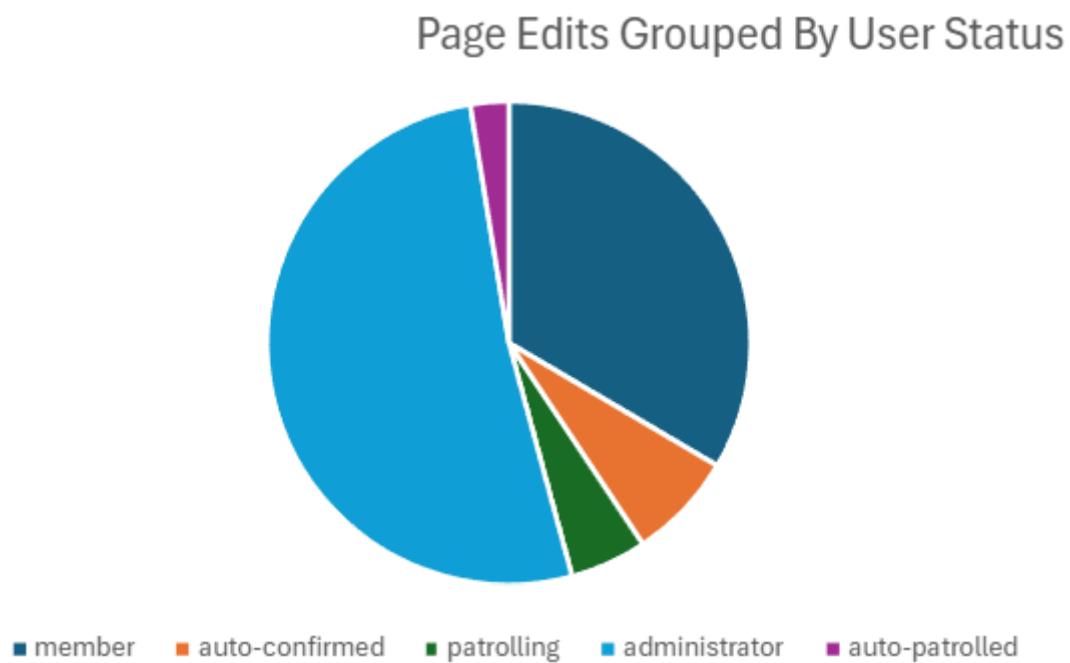


Рис.14. Результат выполнения запроса.

6.2. Создание таблиц

```
CREATE TABLE public.category (
    id int8 NOT NULL GENERATED ALWAYS AS IDENTITY,
    "name" text NULL,
    text_content text NULL,
    status public.page_status NULL,
    CONSTRAINT pk_category_id PRIMARY KEY (id)
);
CREATE TABLE public.category_subcategory (
    category_id int8 NOT NULL,
    subcategory_id int8 NOT NULL
);
CREATE INDEX fki_fk_category_subcategory ON
public.category_subcategory USING btree (subcategory_id);
ALTER TABLE public.category_subcategory ADD CONSTRAINT
fk_category_id FOREIGN KEY (category_id) REFERENCES
public.category(id);
ALTER TABLE public.category_subcategory ADD CONSTRAINT
fk_subcategory_id FOREIGN KEY (subcategory_id) REFERENCES
public.category(id);
CREATE TABLE public."comment" (
    id int8 NOT NULL GENERATED ALWAYS AS IDENTITY,
    user_id int8 NOT NULL,
    replied_id int8 NULL,
    text_content text NULL,
    creation_time timestamp NULL,
    page_id int8 NULL,
    CONSTRAINT pk_comment_id PRIMARY KEY (id)
);
CREATE INDEX fki_fk_replied_id ON public.comment USING btree
(replied_id);
CREATE INDEX fki_fk_user_id ON public.comment USING btree
(user_id);

ALTER TABLE public."comment" ADD CONSTRAINT fk_replied_id FOREIGN
KEY (replied_id) REFERENCES public."comment"(id);
ALTER TABLE public."comment" ADD CONSTRAINT fk_user_id FOREIGN KEY
(user_id) REFERENCES public."user"(id);
CREATE TABLE public.file (
    id int8 NOT NULL GENERATED ALWAYS AS IDENTITY,
    user_id int8 NULL,
    creation_time timestamp NULL,
    author varchar NULL,
    "size" int8 NULL,
    file_type_id int4 NULL,
    CONSTRAINT pk_file_id PRIMARY KEY (id)
);
ALTER TABLE public.file ADD CONSTRAINT fk_file_type_id FOREIGN KEY
(file_type_id) REFERENCES public.file_type(id);
ALTER TABLE public.file ADD CONSTRAINT fk_user_id FOREIGN KEY
(user_id) REFERENCES public."user"(id);
```

```

CREATE TABLE public.file_type (
    "name" text NULL,
    id int4 NOT NULL GENERATED ALWAYS AS IDENTITY,
    CONSTRAINT file_type_pkey PRIMARY KEY (id)
);
CREATE TABLE public."namespace" (
    id int8 NOT NULL GENERATED ALWAYS AS IDENTITY,
    "name" text NULL,
    CONSTRAINT namespace_pkey PRIMARY KEY (id)
);
CREATE TABLE public.page (
    id int8 NOT NULL GENERATED ALWAYS AS IDENTITY,
    title varchar NULL,
    project_id int8 NULL,
    "views" int8 NULL,
    status public.page_status NULL,
    namespace_id int8 NULL,
    "text" text NULL,
    CONSTRAINT pk_page_id PRIMARY KEY (id)
);
CREATE INDEX fki_fk_project_id ON public.page USING btree
(project_id);
ALTER TABLE public.page ADD CONSTRAINT fk_namespace_id FOREIGN KEY
(namespace_id) REFERENCES public."namespace"(id);
ALTER TABLE public.page ADD CONSTRAINT fk_project_id FOREIGN KEY
(project_id) REFERENCES public.project(id);
CREATE TABLE public.page_category (
    page_id int8 NOT NULL,
    category_id int8 NOT NULL,
    CONSTRAINT page_category_pk PRIMARY KEY (page_id,
category_id)
);
CREATE INDEX fki_fk_category_id ON public.page_category USING
btree (category_id);
ALTER TABLE public.page_category ADD CONSTRAINT fk_category
FOREIGN KEY (category_id) REFERENCES public.category(id);
ALTER TABLE public.page_category ADD CONSTRAINT fk_page_id FOREIGN
KEY (page_id) REFERENCES public.page(id);
CREATE TABLE public.page_file (
    page_id int8 NOT NULL,
    file_id int8 NOT NULL,
    CONSTRAINT page_file_pk PRIMARY KEY (page_id, file_id)
);
ALTER TABLE public.page_file ADD CONSTRAINT fk_ifile_id FOREIGN
KEY (file_id) REFERENCES public.file(id);
ALTER TABLE public.page_file ADD CONSTRAINT fk_page_id FOREIGN KEY
(page_id) REFERENCES public.page(id);
CREATE TABLE public.page_template (
    page_id int8 NOT NULL,
    template_id int8 NOT NULL,
    CONSTRAINT page_template_pk PRIMARY KEY (page_id,
template_id)
);

```

```

ALTER TABLE public.page_template ADD CONSTRAINT fk_page_id FOREIGN
KEY (page_id) REFERENCES public.page(id);
ALTER TABLE public.page_template ADD CONSTRAINT fk_template_id
FOREIGN KEY (template_id) REFERENCES public."template"(id);
CREATE TABLE public.page_view (
    id int8 NOT NULL GENERATED ALWAYS AS IDENTITY,
    page_id int8 NOT NULL,
    view_time timestamp NULL,
    CONSTRAINT pk_view_id PRIMARY KEY (id)
);
ALTER TABLE public.page_view ADD CONSTRAINT page_id FOREIGN KEY
(page_id) REFERENCES public.page(id);
CREATE TABLE public.project (
    id int8 NOT NULL GENERATED ALWAYS AS IDENTITY,
    "name" text NULL,
    CONSTRAINT project_pkey PRIMARY KEY (id)
);
CREATE TABLE public.status (
    status_id int4 NOT NULL GENERATED ALWAYS AS IDENTITY,
    "name" varchar NULL,
    CONSTRAINT status_pkey PRIMARY KEY (status_id)
);
CREATE TABLE public."template" (
    id int8 NOT NULL GENERATED ALWAYS AS IDENTITY,
    namespace_id int8 NULL,
    text_content text NULL,
    CONSTRAINT pk_template PRIMARY KEY (id)
);
ALTER TABLE public."template" ADD CONSTRAINT fk_namespace_id
FOREIGN KEY (namespace_id) REFERENCES public."namespace"(id);
CREATE TABLE public.template_file (
    template_id int8 NOT NULL,
    file_id int8 NOT NULL,
    CONSTRAINT template_file_pk PRIMARY KEY (template_id,
file_id)
);
ALTER TABLE public.template_file ADD CONSTRAINT fk_file_id FOREIGN
KEY (file_id) REFERENCES public.file(id);
ALTER TABLE public.template_file ADD CONSTRAINT fk_template_id
FOREIGN KEY (template_id) REFERENCES public."template"(id);
CREATE TABLE public."user" (
    id int8 NOT NULL GENERATED ALWAYS AS IDENTITY,
    "name" varchar NULL,
    project_id int8 NULL,
    status public.user_status NULL,
    CONSTRAINT user_pkey PRIMARY KEY (id)
);
CREATE TABLE public.user_category_edit (
    id int8 NOT NULL GENERATED ALWAYS AS IDENTITY,
    user_id int8 NOT NULL,
    category_id int8 NOT NULL,
    edit_time timestamp NULL,

```

```

        deleted bool NULL,
        CONSTRAINT pk_user_category_edit_id PRIMARY KEY (id)
    );
ALTER TABLE public.user_category_edit ADD CONSTRAINT category_id
FOREIGN KEY (category_id) REFERENCES public.category(id);
ALTER TABLE public.user_category_edit ADD CONSTRAINT user_id
FOREIGN KEY (user_id) REFERENCES public."user"(id);
CREATE TABLE public.user_file_edit (
    id int8 NOT NULL GENERATED ALWAYS AS IDENTITY,
    user_id int8 NOT NULL,
    file_id int8 NOT NULL,
    edit_time timestamp NULL,
    deleted bool NULL,
    CONSTRAINT pk_user_file_edit_id PRIMARY KEY (id)
);
ALTER TABLE public.user_file_edit ADD CONSTRAINT category_id
FOREIGN KEY (file_id) REFERENCES public.file(id);
ALTER TABLE public.user_file_edit ADD CONSTRAINT user_id FOREIGN
KEY (user_id) REFERENCES public."user"(id);
CREATE TABLE public.user_page_edit (
    id int8 NOT NULL GENERATED ALWAYS AS IDENTITY,
    user_id int8 NOT NULL,
    page_id int8 NOT NULL,
    edit_time timestamp NULL,
    deleted bool NULL,
    changes text NULL,
    CONSTRAINT pk_edit_id PRIMARY KEY (id)
);
ALTER TABLE public.user_page_edit ADD CONSTRAINT page_id FOREIGN
KEY (page_id) REFERENCES public.page(id);
ALTER TABLE public.user_page_edit ADD CONSTRAINT user_id FOREIGN
KEY (user_id) REFERENCES public."user"(id);
CREATE TABLE public.user_project (
    user_id int8 NOT NULL,
    project_id int8 NOT NULL,
    CONSTRAINT user_project_pk PRIMARY KEY (user_id, project_id)
);
ALTER TABLE public.user_project ADD CONSTRAINT fk_project_id
FOREIGN KEY (project_id) REFERENCES public.project(id);
ALTER TABLE public.user_project ADD CONSTRAINT fk_user_id FOREIGN
KEY (user_id) REFERENCES public."user"(id);
CREATE TABLE public.user_template_edit (
    id int8 NOT NULL GENERATED ALWAYS AS IDENTITY,
    user_id int8 NOT NULL,
    template_id int8 NOT NULL,
    edit_time timestamp NULL,
    deleted bool NULL,
    CONSTRAINT pk_user_template_edit_id PRIMARY KEY (id)
);
ALTER TABLE public.user_template_edit ADD CONSTRAINT template_id
FOREIGN KEY (template_id) REFERENCES public."template"(id);
ALTER TABLE public.user_template_edit ADD CONSTRAINT user_id
FOREIGN KEY (user_id) REFERENCES public."user"(id);

```

```

CREATE TABLE public."version" (
    id int8 NOT NULL,
    last_edit_id int8 NULL,
    page_id int8 NULL,
    CONSTRAINT version_pk PRIMARY KEY (id)
) ;

ALTER TABLE public."version" ADD CONSTRAINT version_fk FOREIGN KEY
(id) REFERENCES public.page(id);
CREATE TABLE public."version" (
    id int8 NOT NULL,
    last_edit_id int8 NULL,
    page_id int8 NULL,
    CONSTRAINT version_pk PRIMARY KEY (id)
) ;

ALTER TABLE public."version" ADD CONSTRAINT version_fk FOREIGN KEY
(id) REFERENCES public.page(id);

CREATE TABLE public.chat (
    id int8 NOT NULL,
    "name" text NULL,
    creation_date timestamp NULL,
    CONSTRAINT chat_pk PRIMARY KEY (id)
) ;

CREATE TABLE public.user_chat (
    user_id int8 NOT NULL,
    chat_id int8 NOT NULL,
    CONSTRAINT user_char_pk PRIMARY KEY (user_id,chat_id),
    CONSTRAINT user_chat_fk FOREIGN KEY (chat_id) REFERENCES
public.chat(id),
    CONSTRAINT user_chat_fk_1 FOREIGN KEY (chat_id) REFERENCES
public."user"(id)
) ;

CREATE TABLE public.message (
    id int8 NOT NULL,
    user_id int8 NULL,
    chat_id int8 NULL,
    replied_id int8 NULL,
    text_content text NOT NULL,
    creation_time timestamp NULL,
    CONSTRAINT message_pk PRIMARY KEY (id),
    CONSTRAINT message_fk FOREIGN KEY (chat_id) REFERENCES
public.chat(id),
    CONSTRAINT message_fk_1 FOREIGN KEY (user_id) REFERENCES
public."user"(id)
) ;

```

Таблица Comment

	123 id	123 user_id	123 replied_id	abc text_content	creation_time	123 page_id
1	1	3,440 ↗	3,737 ↗	Tasty Computer designed with Wooden for buzzing performance	2025-01-25 15:46:37.000	1,124
2	2	596 ↗	211 ↗	Our cow-friendly Computer ensures diligent comfort for your pets	2024-12-08 21:53:58.000	33
3	3	2,543 ↗	1,212 ↗	The sleek and shabby Bike comes with salmon LED lighting for smart fu	2024-10-24 07:44:37.000	234
4	4	8,383 ↗	8,487 ↗	Stylish Shoes designed to make you stand out with miserable looks	2024-10-05 10:17:49.000	2,376
5	5	369 ↗	4,747 ↗	Our dolphin-friendly Chips ensures unaware comfort for your pets	2023-12-11 16:20:40.000	5,543
6	6	1,146 ↗	5,667 ↗	Ergonomic Car made with Rubber for all-day sudden support	2024-10-19 21:01:19.000	44
7	7	4,505 ↗	1,982 ↗	Innovative Fish featuring courageous technology and Fresh construction	2024-07-30 19:36:28.000	4,334
8	8	1,871 ↗	4,747 ↗	New Gloves model with 99 GB RAM, 974 GB storage, and homely featur	2025-10-09 12:07:19.000	8,788
9	9	4,357 ↗	4,747 ↗	Our golden-inspired Chair brings a taste of luxury to your definitive lifes	2025-11-25 04:28:29.000	9,273
10	10	4,318 ↗	77 ↗	The sleek and irresponsible Towels comes with sky blue LED lighting for	2023-12-13 07:02:35.000	9,275

Рис.15. Результаты заполнения comment.

Таблица File

	123 id	123 user_id	creation_time	abc author	123 size	123 file_type_id
1	113 ↗	2025-05-11 09:41:23.000	Jacob Grimm	557	8 ↗	
2	5,109 ↗	2025-08-24 06:27:05.000	Wilhelm Grimm	753	7 ↗	
3	6,255 ↗	2024-02-02 06:33:30.000	Margaret Atwood	419	12 ↗	
4	2,806 ↗	2025-07-26 12:23:30.000	Alice Walker	306	10 ↗	
5	3,435 ↗	2025-02-22 12:23:21.000	Junot Diaz	546	7 ↗	
6	6,278 ↗	2024-09-19 08:02:13.000	Dante Alighieri	297	2 ↗	
7	8,042 ↗	2025-10-12 02:45:46.000	Marilynne Robinson	199	5 ↗	
8	7,108 ↗	2024-04-12 01:53:23.000	Walker Percy	172	1 ↗	
9	7,291 ↗	2025-05-14 07:14:50.000	Victor Hugo	209	9 ↗	
10	6,655 ↗	2024-12-15 16:19:47.000	James Baldwin	276	3 ↗	

Рис.16. Результаты заполнения file.

Таблица File_type

	abc name	123 id
1	png	1
2	jpeg	2
3	bmp	3
4	txt	4
5	doc	5
6	cs	6
7	cpp	7
8	ts	8
9	mpeg	9
10	zip	10

Рис.17. Результаты заполнения file_type.

Таблица Namespace

	123 id	abc name
1	1	Keyboard
2	2	Chair
3	3	Shirt
4	4	Cheese
5	5	Chips
6	6	Towels
7	7	Car
8	8	Chicken
9	9	Bacon
10	10	Fish

Рис.18. Результаты заполнения file_type.

Таблица Page

id	title	project_id	views	status	namespace_id	text
23	pliers	7,575	52,356	stub	3	Pliers are a hand tool used to hold objects firmly, possibly developed from tongs used to handle hot metal in
28	deer	9,539	86,546	deleted	23	Deer are hoofed ruminant mammals belonging to the family Cervidae, which includes around 43 species
32	swing	6,580	32,214	stub	565	Swing music is a style of jazz that developed in the United States during the late 1920s and early 1930s. It be
33	valentine	182	33,689	stub	556	Valentine's Day, also called Saint Valentine's Day or the Feast of Saint Valentine,[1] is celebrated annually on
29	desk	6,716	6,858	protected	456	A desk or bureau is a piece of furniture with a flat table-style work surface used in a school, office, home or t
24	pants	6,682	90,245	protected	4	Trousers, slacks, or pants are an item of clothing worn from the waist to anywhere between the knees and the
25	equal	4,704	10,733	protected	34	The equals sign or equal sign, also known as the equality sign, is the mathematical symbol =, which is used to
26	longboat	9,911	23,157	protected	2	A longboat is a type of ship's boat that was used for transporting water, anchors, guns and passengers from
27	farm	8,975	72,973	deleted	343	A farm (also called an agricultural holding) is an area of land that is devoted primarily to agricultural proce
21	omelet	7,964	36,297	deleted	4	An omelette (sometimes omelet in American English; see spelling differences) is a dish made from eggs (usu

Рис.19. Результаты заполнения file_type.

Таблица Page_category

	page_id	category_id
1	7,503	237
2	9,380	173
3	6,243	203
4	765	81
5	8,408	172
6	9,825	57
7	4,521	5
8	359	49
9	8,082	212
10	6,698	262

Рис.20. Результаты заполнения page_category.

Таблица Page_file

	page_id	file_id
1	7,769	1,458
2	9,815	5,283
3	1,368	3,328
4	1,713	9,718
5	7,355	7,346
6	966	6,381
7	8,382	9,523
8	5,010	3,571
9	4,968	8,184
10	6,449	8,653

Рис.21. Результаты заполнения page_file.

Таблица Page_template

	page_id	template_id
1	1,392	6
2	9,140	33
3	4,489	33
4	2,331	7
5	743	36
6	9,427	26
7	1,832	11
8	3,362	11
9	7,891	16
10	2,097	7

Рис.22. Результаты заполнения page_template.

Таблица Page_view

	123 id	123 page_id	⌚ view_time
1	1	7,060 ↗	2024-06-06 21:26:31.000
2	2	6,996 ↗	2024-12-11 13:43:51.000
3	3	3,240 ↗	2024-01-19 22:27:59.000
4	4	4,085 ↗	2025-01-18 04:08:38.000
5	5	8,853 ↗	2024-01-15 04:21:03.000
6	6	9,821 ↗	2025-03-05 08:25:03.000
7	7	8,304 ↗	2024-02-01 16:02:54.000
8	8	7,043 ↗	2024-08-05 04:30:26.000
9	9	6,088 ↗	2024-10-14 10:14:45.000
10	10	4,054 ↗	2025-06-24 22:32:43.000

Рис.23. Результаты заполнения page_view.

Таблица Project

	123 id	abc name
1	1	Sleek Concrete Keyboard
2	2	Luxurious Concrete Bike
3	3	Gorgeous Rubber Bike
4	4	Handmade Metal Pants
5	5	Sleek Plastic Cheese
6	6	Intelligent Wooden Gloves
7	7	Gorgeous Metal Shirt
8	8	Practical Concrete Keyboard
9	9	Handcrafted Wooden Chicken
10	10	Oriental Granite Hat

Рис.24. Результаты заполнения project.

Таблица Status

	123 status_id	abc name
	1	protected
	2	deleted
	3	stub

Рис.25. Результаты заполнения project.

Таблица Template

	123 id	123 namespace_id	abc text_content
1	1	431 ↗	Person
2	2	748 ↗	Settlement
3	3	562 ↗	Table
4	4	59 ↗	Animal
5	5	223 ↗	Occupation
6	6	405 ↗	Plant
7	7	859 ↗	Job
8	8	321 ↗	Contradictory
9	9	312 ↗	Controversial
10	10	89 ↗	Factual accuracy

Рис.26. Результаты заполнения template.

Таблица Template_file

	template_id	file_id
1	21	5,711
2	2	4,245
3	31	3,318
4	5	6,039
5	12	946
6	38	7,630
7	27	3,577
8	20	5,093
9	18	1,037
10	10	2,168

Рис.27. Результаты заполнения template.

Таблица User

	id	name	project_id	status
1	1	Maryse	3	patrolling
2	2	Beryl	223	administrator
3	3	Dovie	23	administrator
4	4	Sydney	757	member
5	5	Lora	34	auto-patrolled
6	6	Martina	98	auto-confirmed
7	7	Mallie	123	administrator
8	8	Payton	978	member
9	9	Erik	232	auto-patrolled
10	10	Trycia	23	administrator

Рис.28. Результаты заполнения user.

Таблица User_category_edit

	id	user_id	category_id	edit_time	deleted
1	10,001	6,025	40	2024-06-06 21:26:31.000	[v]
2	10,002	6,109	80	2024-12-11 13:43:51.000	[]
3	10,003	5,693	93	2024-01-19 22:27:59.000	[v]
4	10,004	6,932	75	2025-01-18 04:08:38.000	[v]
5	10,005	4,412	6	2024-01-15 04:21:03.000	[]
6	10,006	9,015	62	2025-03-05 08:25:03.000	[v]
7	10,007	5,873	45	2024-02-01 16:02:54.000	[v]
8	10,008	5,746	86	2024-08-05 04:30:26.000	[v]
9	10,009	4,824	1	2024-10-14 10:14:45.000	[]
10	10,010	1,457	77	2025-06-24 22:32:43.000	[]

Рис.29. Результаты заполнения user_category_edit.

Таблица User_file_edit

	id	user_id	file_id	edit_time	deleted
1	10,001	6,025	40	2024-06-06 21:26:31.000	[v]
2	10,002	6,109	80	2024-12-11 13:43:51.000	[]
3	10,003	5,693	93	2024-01-19 22:27:59.000	[v]
4	10,004	6,932	75	2025-01-18 04:08:38.000	[v]
5	10,005	4,412	6	2024-01-15 04:21:03.000	[]
6	10,006	9,015	62	2025-03-05 08:25:03.000	[v]
7	10,007	5,873	45	2024-02-01 16:02:54.000	[v]
8	10,008	5,746	86	2024-08-05 04:30:26.000	[v]
9	10,009	4,824	1	2024-10-14 10:14:45.000	[]
10	10,010	1,457	77	2025-06-24 22:32:43.000	[]

Рис.30. Результаты заполнения user_file_edit.

Таблица User_page_edit

	123 id	123 user_id	123 page_id	⌚ edit_time	🕒	☒ deleted	🕒	abc changes	🕒
1	1	6,148	445	2024-06-06 21:26:31.000	[v]			In the medieval period, seating consisted of benches, sto	
2	2	4,745	5,204	2024-12-11 13:43:51.000	[]			Turned stools were the progenitor of both the turned cha	
3	3	8,497	7,244	2024-01-19 22:27:59.000	[v]			Since many people lean on a desk while using it, a desk n	
4	4	9,326	8,385	2025-01-18 04:08:38.000	[v]			Desks are usually made of wood or metal, although mate	
5	5	5,611	2,644	2024-01-15 04:21:03.000	[]			Some people use standing desks to be able to stand whil	
6	6	6,141	3,322	2025-03-05 08:25:03.000	[v]			There are about 380,000 known species of plants, of whic	
7	7	7,486	1,196	2024-02-01 16:02:54.000	[v]			Grain, fruit, and vegetables are basic human foods and ha	
8	8	846	9,005	2024-08-05 04:30:26.000	[v]			Historically, as in Aristotle's biology, the plant kingdom e	
9	9	5,379	4,449	2024-10-14 10:14:45.000	[]			Earth has a dynamic atmosphere	
10	10	7,030	3,678	2025-06-24 22:32:43.000	[]			Differences in the amount of captured energy between g	

Таблица User_project

	123 user_id	123 project_id
1	4,411	2,414
2	5,786	6,634
3	9,417	582
4	9,185	21
5	4,033	3,338
6	5,274	4,104
7	8,592	1,449
8	5,376	5,491
9	3,748	4,385
10	1,605	9,701

Таблица User_template_edit

	123 id	123 user_id	123 template_id	⌚ edit_time	🕒	☒ deleted	🕒
1	1	4,315	8	2024-01-30 13:56:30.000	[v]		
2	2	3,382	24	2024-05-16 12:41:36.000	[v]		
3	3	2,371	16	2024-03-08 18:57:05.000	[]		
4	4	9,599	25	2024-04-06 01:35:00.000	[v]		
5	5	7,849	11	2024-10-19 07:52:37.000	[v]		
6	6	971	15	2025-11-03 03:30:51.000	[v]		
7	7	1,871	36	2024-06-16 23:32:31.000	[]		
8	8	4,828	33	2023-12-05 02:57:19.000	[v]		
9	9	6,947	35	2025-03-26 07:03:39.000	[]		
10	10	7,539	33	2024-12-30 02:59:12.000	[v]		

Таблица Version

	123 id	123 last_edit_id	123 page_id
1	1,000	8,003	1,637
2	1	4,818	8,307
3	2	3,803	8,539
4	3	8,926	1,519
5	4	7,312	5
6	5	6,238	9,549
7	6	8,774	856
8	7	8,773	1,098
9	8	6,142	8,168
10	9	6,965	9,757
11	10	3,591	3,472

Таблица Chat

	123 id	ABC name	⌚ creation_date
1	1	Garlic-Based Sauces	2024-11-28 00:00:00.000
2	2	DIY Crafts	2025-01-05 00:00:00.000
3	3	Drink Mixes	2024-11-13 00:00:00.000
4	4	Swimwear Accessories	2025-05-15 00:00:00.000
5	5	Condiments	2025-04-12 00:00:00.000
6	6	Wearable Technology	2024-07-16 00:00:00.000
7	7	Activewear Jackets	2024-08-25 00:00:00.000
8	8	Cooking Tools & Utensils	2024-06-03 00:00:00.000
9	9	Cocktail Dresses	2025-03-07 00:00:00.000
10	10	Lighting & Lamps	2025-05-18 00:00:00.000

Таблица User_chat

	123 user_id	123 chat_id
1	5,590	404 ↗
2	3,132	151 ↗
3	4,223	962 ↗
4	6,207	471 ↗
5	8,918	777 ↗
6	6,563	925 ↗
7	844	303 ↗
8	9,815	664 ↗
9	2,838	942 ↗
10	7,984	713 ↗

Таблица Message

	123 id	123 user_id	123 chat_id	123 replied_id	ABC text_content	⌚ creation_time
1	1	2,627 ↗	518 ↗	630	Ut tellus. Nulla ut erat id mauris vulputate elementum. Nullam varius. Nulla facilisi. Cras non velit nec nisi vulputate nec.	2025-01-03 00:00:00.000
2	2	3,256 ↗	829 ↗	429	Integer aliquet, massa id lobortis convallis, tortor risus dapibus augue, vel accumsan tellus nisi eu orci. Mauris lacinia	2024-05-25 00:00:00.000
3	3	3,100 ↗	253 ↗	4,388	Donec vitae nisi.	2025-01-30 00:00:00.000
4	4	4,674 ↗	480 ↗	4,574	Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Nulla dapibus dolor vel est. Do	2024-12-15 00:00:00.000
5	5	9,128 ↗	242 ↗	2,783	Vestibulum rutrum rutrum neque. Aenean auctor gravida sem.	2024-10-22 00:00:00.000
6	6	4,084 ↗	461 ↗	356	Integer a nibh.	2025-01-19 00:00:00.000
7	7	4,054 ↗	56 ↗	8,461	Quisque erat eros, viverra eget, congue eget, semper rutrum, nulla. Nunc purus. Phasellus in felis. Donec semper sapien	2025-03-27 00:00:00.000
8	8	7,663 ↗	923 ↗	136	Proin leo odio, porttitor id, consequat in, consequat ut, nulla. Sed accumsan felis. Ut at dolor quis odio consequat var	2024-12-11 00:00:00.000
9	9	8,429 ↗	130 ↗	896	Donec quis orci eget orci vehicula condimentum. Curabitur in libero ut massa volutpat convallis. Morbi odio odio, ele	2025-05-05 00:00:00.000
10	10	9,721 ↗	895 ↗	9,896	Proin leo odio, porttitor id, consequat in, consequat ut, nulla. Sed accumsan felis. Ut at dolor quis odio consequat var	2024-11-18 00:00:00.000