

# ОТЧЁТ

Лабораторная работа №2-1:  
«Пользователи. Роли. Привилегии»

Группа  
Студент  
Преподаватель

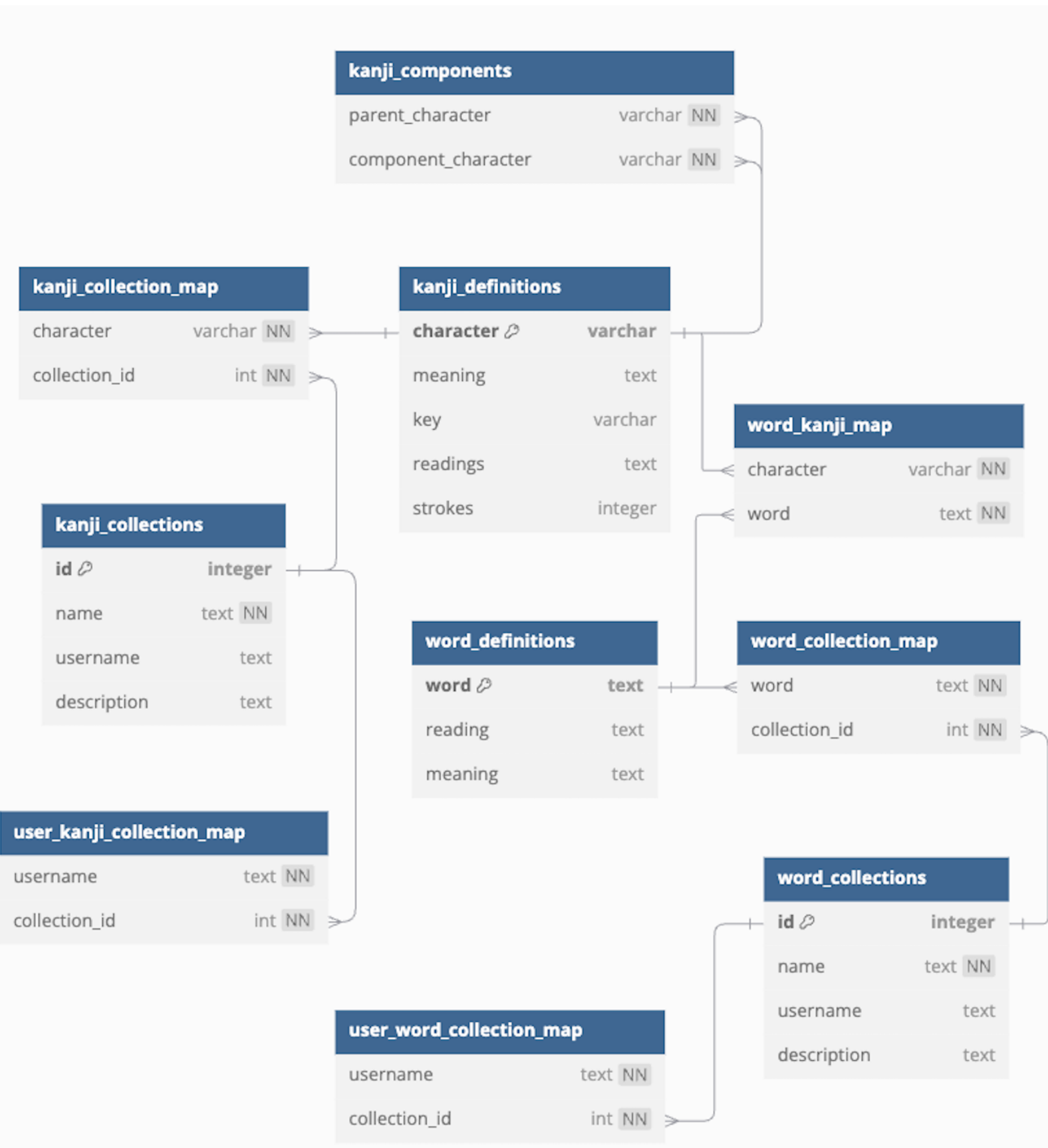
Б21-525  
Р.Т. Мясников  
М.А. Куприяшин

# Оглавление

1.	Диаграмма отношений сущностей . . . . .	3
2.	Схемы . . . . .	4
3.	Роли . . . . .	6
4.	Заключение . . . . .	12
5.	Приложение . . . . .	13

# 1. Диаграмма отношений сущностей

Стоит отметить, что на данной диаграмме используются только отношения 0..n.



## 2. Схемы

По умолчанию все таблицы в базе данных находились в схеме `public`. Для проверки этого был использован запрос:

```
SELECT table_schema, table_name
FROM information_schema.tables
WHERE table_type = 'BASE TABLE' AND table_schema NOT IN ('
    pg_catalog', 'information_schema');
```

### Результат запроса

table_schema	table_name
public	word_collections
public	word_collection_map
public	kanji_collections
public	user_word_collection_map
public	kanji_definitions
public	word_kanji_map
public	word_definitions
public	kanji_components
public	user_kanji_collection_map
public	kanji_collection_map

Чтобы не предоставлять лишние привелегии при работе с базой данных, следует изменить схему всем таблицам. Кроме того, для структурирования данных и разграничения пространства имён, таблицы были разделены на данные схемы: `kanji`, `word`, `user_collections`, а названия таблиц были изменены:

table_schema	table_name
kanji	components
user_collections	word
user_collections	word_map
user_collections	kanji
user_collections	user_word_collection_map
kanji	definitions
word	kanji_map
word	definitions
user_collections	user_kanji_collection_map
user_collections	kanji_map

### 3. Роли

Для функционирования базы данных были созданы следующие роли:

- **admin** (администратор базы данных) — роль для управления базой данных и правами. Из системных привелегий этой роли понадобятся следующие права: **LOGIN**, **CREATEDB**, **CREATEROLE**. Из объектных прав понадобятся все, с возможностью передачи их другим ролям;
- **privileged\_developer** — роль для повышения привелегий разработчику. Кроме привелегий самого разработчика включает **CREATEDB** и **TRUNCATE** для таблиц;
- **developer** (разработчик) — роль для работы с таблицами. Этой роли потребуется доступ к изменению структуры базы данных, но доступ к управлению правами должен быть ограничен. Системные привелегии: **LOGIN**, **CREATEDB**, **NOINHERIT**. Из объектных прав понадобятся все, но без возможности передачи прав. Роль должна наследоваться от **privileged\_developer**;
- **content\_creator** — роль для пользователя, который заполняет данными таблицы из схем **kanji**, **word**, **user\_collections**. Системные привелегии: **LOGIN**, **NOINHERIT**. Этой роли потребуется разрешение на изменение таблиц схем: **SELECT**, **INSERT**, **UPDATE**, **DELETE**. И использования схем: **USAGE** и использование **SEQUENCES** для **user\_collections**: **SELECT**, **USAGE**;
- **content\_user** (пользователь) — роль для просмотра данных и изменения пользовательских списков. Этой роли потребуется разрешение на **SELECT** схем **kanji** и **verbs**, а также доступ к изменению таблиц из схемы **user\_collections**: **SELECT**, **INSERT**, **UPDATE**, **DELETE**. А также разрешение использовать схемы **kanji**, **word**, **user\_collections**: **USAGE**; и использование **SEQUENCES** для **user\_collections**: **SELECT**, **USAGE**;

#### Скрипт

```
CREATE ROLE admin WITH LOGIN CREATEDB CREATEROLE PASSWORD 'admin';
CREATE ROLE privileged_developer WITH CREATEDB NOINHERIT PASSWORD '
sudo';
CREATE ROLE developer WITH LOGIN NOINHERIT PASSWORD 'developer';
CREATE ROLE content_creator WITH LOGIN NOINHERIT PASSWORD '
content_creator';
CREATE ROLE content_user WITH LOGIN NOINHERIT PASSWORD '
content_user';

-- admin
GRANT ALL PRIVILEGES ON DATABASE japanese_db TO admin WITH GRANT
OPTION;
```

```

DO $$
DECLARE
    schema TEXT;
    schemas_array TEXT[] := ARRAY['kanji', 'word', '
        user_collections'];
BEGIN
    FOREACH schema IN ARRAY schemas_array
    LOOP
        EXECUTE format('GRANT ALL PRIVILEGES ON ALL TABLES IN
            SCHEMA %I TO admin WITH GRANT OPTION;', schema);
        EXECUTE format('GRANT ALL PRIVILEGES ON ALL SEQUENCES IN
            SCHEMA %I TO admin WITH GRANT OPTION;', schema);
        EXECUTE format('GRANT ALL PRIVILEGES ON ALL FUNCTIONS IN
            SCHEMA %I TO admin WITH GRANT OPTION;', schema);
        EXECUTE format('GRANT ALL PRIVILEGES ON SCHEMA %I TO admin
            WITH GRANT OPTION;', schema);
    END LOOP;
END $$;

-- privileged_developer
GRANT ALL PRIVILEGES ON DATABASE japanese_db TO
    privileged_developer;

DO $$
DECLARE
    schema TEXT;
    schemas_array TEXT[] := ARRAY['kanji', 'word', '
        user_collections'];
BEGIN
    FOREACH schema IN ARRAY schemas_array
    LOOP
        EXECUTE format('GRANT ALL PRIVILEGES ON ALL TABLES IN
            SCHEMA %I TO privileged_developer;', schema);
        EXECUTE format('GRANT ALL PRIVILEGES ON ALL SEQUENCES IN
            SCHEMA %I TO privileged_developer;', schema);
        EXECUTE format('GRANT ALL PRIVILEGES ON ALL FUNCTIONS IN
            SCHEMA %I TO privileged_developer;', schema);
        EXECUTE format('GRANT ALL PRIVILEGES ON SCHEMA %I TO
            privileged_developer;', schema);
    END LOOP;
END $$;

-- developer
GRANT ALL PRIVILEGES ON DATABASE japanese_db TO developer;

DO $$
DECLARE
    schema TEXT;
    schemas_array TEXT[] := ARRAY['kanji', 'word', '
        user_collections'];
BEGIN
    FOREACH schema IN ARRAY schemas_array
    LOOP
        EXECUTE format('GRANT SELECT, INSERT, UPDATE, DELETE,
            REFERENCES, TRIGGER ON ALL TABLES IN SCHEMA %I TO
            developer;', schema);
        EXECUTE format('GRANT ALL PRIVILEGES ON ALL SEQUENCES IN
            SCHEMA %I TO developer;', schema);
        EXECUTE format('GRANT ALL PRIVILEGES ON ALL FUNCTIONS IN
            SCHEMA %I TO developer;', schema);
    END LOOP;
END $$;

```

```

        EXECUTE format('GRANT ALL PRIVILEGES ON SCHEMA %I TO
                        developer;', schema);
    END LOOP;
END $$;

-- content_creator
GRANT CONNECT ON DATABASE japanese_db TO content_creator;

GRANT SELECT, INSERT, UPDATE, DELETE ON ALL TABLES IN SCHEMA kanji
TO content_creator;
GRANT SELECT, INSERT, UPDATE, DELETE ON ALL TABLES IN SCHEMA word
TO content_creator;
GRANT SELECT, INSERT, UPDATE, DELETE ON ALL TABLES IN SCHEMA
user_collections TO content_creator;

GRANT SELECT, USAGE ON ALL SEQUENCES IN SCHEMA user_collections TO
content_creator;

GRANT USAGE ON SCHEMA kanji TO content_creator;
GRANT USAGE ON SCHEMA word TO content_creator;
GRANT USAGE ON SCHEMA user_collections TO content_creator;

-- content_user
GRANT CONNECT ON DATABASE japanese_db TO content_user;

GRANT SELECT ON ALL TABLES IN SCHEMA kanji TO content_user;
GRANT SELECT ON ALL TABLES IN SCHEMA word TO content_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON ALL TABLES IN SCHEMA
user_collections TO content_user;

GRANT SELECT, USAGE ON ALL SEQUENCES IN SCHEMA user_collections TO
content_user;

GRANT USAGE ON SCHEMA kanji TO content_user;
GRANT USAGE ON SCHEMA word TO content_user;
GRANT USAGE ON SCHEMA user_collections TO content_user;

GRANT privileged_developer TO developer;

```

## Проверка системных привелегий

```

japanese_db=> SELECT rolname, rolsuper, rolinherit, rolcreatorole, rolcreatedb, rolcanlogin
FROM pg_catalog.pg_roles
WHERE rolname IN ('admin', 'privileged_developer', 'developer', 'content_user', 'content_creator');

```

rolname	rolsuper	rolinherit	rolcreatorole	rolcreatedb	rolcanlogin
admin	f	t	t	t	t
privileged_developer	f	f	f	t	f
developer	f	f	f	f	t
content_creator	f	f	f	f	t
content_user	f	f	f	f	t

(5 rows)



## Проверка объектных привелегий

Для content\_user проверка доступа к чтению таблиц из kanji, word, user\_collections:

```
~/Projects/Mephi/DB/mephi_database_systems/lab2_1 git:(main)±2
psql -h localhost -p 5432 -U content_user japanese_db

japanese_db=> SELECT DISTINCT wk.character
japanese_db-> FROM user_collections.word_map AS cw_map
japanese_db-> JOIN word.kanji_map AS wk ON cw_map.word = wk.word
japanese_db-> WHERE cw_map.collection_id = 1;
 character
-----
 火
 土
 曜
 月
 水
 金
 日
 木
(8 rows)

japanese_db=> SELECT character, meaning, readings
japanese_db-> FROM kanji.definitions
japanese_db-> ORDER BY strokes;
 character | meaning | readings
-----|-----|-----
 一         | one     | ひと,イチ
 八         | eight   | や,ハチ,ハツ
 七         | seven   | なな,なな.つ,なの,シチ
 人         | person  | ひと,ジン,ニン
 九         | nine    | この,この.つ,キュウ,ク
 力         | power   | ちから,リョク,リキ,リイ
 二         | two     | ふた,ニ,ジ
 十         | ten     | とお,と,ジュウ,ジツ,ジュツ
 入         | enter,insert | い.る,い.れる,ニュウ,ジュ
 大         | big     | おお.きい,ダイ,タイ
 子         | child   | こ,シ
 土         | soil,earth | つち,ド,ト
 小         | small   | ちい.さい,ショウ
```

Для content\_user проверка доступа на изменение таблиц в user\_collections и запрет на изменение kanji и word:

```
~/Projects/Mephi/DB/mephi_database_systems/lab2_1 git:(main)±2
psql -h localhost -p 5432 -U content_user japanese_db

Password for user content_user:
psql (17.0)
Type "help" for help.

japanese_db=> INSERT INTO user_collections.kanji (name, username)
VALUES ('furniture', 'bob');
INSERT 0 1
japanese_db=>

japanese_db=> DELETE FROM user_collections.word_map
WHERE collection_id=1 AND word='月曜日';
DELETE 1
japanese_db=> UPDATE user_collections.kanji
SET username = 'newUser'
WHERE id = 4;
UPDATE 1
japanese_db=>

japanese_db=> UPDATE kanji.definitions
SET meaning='Hahaa'
WHERE character='人';
ERROR: permission denied for table definitions
japanese_db=> DELETE FROM kanji.definitions
WHERE character='人';
ERROR: permission denied for table definitions
japanese_db=> UPDATE word.definitions
SET meaning='Hahaa'
WHERE word='今日';
ERROR: permission denied for table definitions
japanese_db=> DELETE FROM word.definitions
WHERE word='今日';
ERROR: permission denied for table definitions
japanese_db=>
```

Для content\_user проверка запрета TRUNCATE:

```
japanese_db=> TRUNCATE word.definitions CASCADE;
ERROR: permission denied for table definitions
japanese_db=> TRUNCATE kanji.definitions CASCADE;
ERROR: permission denied for table definitions
japanese_db=> TRUNCATE user_collections.definitions CASCADE;
ERROR: schema "user_collections" does not exist
japanese_db=> TRUNCATE user_collections.definitions CASCADE;
ERROR: relation "user_collections.definitions" does not exist
japanese_db=> TRUNCATE user_collections.kanji CASCADE;
ERROR: permission denied for table kanji
japanese_db=>
```

Для content\_creator права идентичные, не считая права на изменение kanji и word:

```
japanese_db=> DELETE FROM word.kanji_map
WHERE word='今日';
DELETE 2
japanese_db=> UPDATE kanji.definitions
SET meaning='Hahaha'
WHERE character='人';
UPDATE 1
japanese_db=> UPDATE word.definitions
SET meaning='Hahaha'
WHERE word='人';
UPDATE 1
japanese_db=> DELETE FROM kanji.components
japanese_db-> WHERE parent_character='今';
DELETE 2
japanese_db=> INSERT INTO kanji.definitions (character, meaning)
japanese_db-> VALUES ('K', 'Hahaaa');
INSERT 0 1
japanese_db=> INSERT INTO word.definitions (word, meaning)
VALUES ('K', 'Hahaaa');
INSERT 0 1
japanese_db=>
```

Для developer изменение и просмотр kanji, word и user\_collections:

```
~/Projects/Mephi/DB/mephi_database_systems/lab2_1 git:(main)±2
psql -h localhost -p 5432 -U developer japanese_db

japanese_db=> INSERT INTO word.definitions (word, meaning)
VALUES ('K', 'Hahaaa');
INSERT 0 1
japanese_db=> INSERT INTO kanji.definitions (character, meaning)
VALUES ('K', 'Hahaaa');
INSERT 0 1
japanese_db=> DELETE FROM kanji.components
WHERE parent_character='今';
DELETE 2
japanese_db=> DELETE FROM word.kanji_map
WHERE word='今日';
DELETE 2
japanese_db=> UPDATE kanji.definitions
SET meaning='Hahaha'
WHERE character='人';
UPDATE 1
japanese_db=> UPDATE word.definitions
SET meaning='Hahaha'
WHERE word='人';
UPDATE 1
japanese_db=> SELECT DISTINCT wk.character
japanese_db-> FROM user_collections.word_map AS cw_map
japanese_db-> JOIN word.kanji_map AS wk ON cw_map.word = wk.word
japanese_db-> WHERE cw_map.collection_id = 1;
character
-----
土
日
曜
月
木
水
火
金
```

Для developer TRUNCATE:

```
~/Projects/Mephi/DB/mephi_database_systems/lab2_1 git:(main)±2
psql -h localhost -p 5432 -U developer japanese_db
日本語
金
(8 rows)

japanese_db=> TRUNCATE kanji.defenitions CASCADE;
ERROR: relation "kanji.defenitions" does not exist
japanese_db=> TRUNCATE TABLE kanji.defenitions CASCADE;
ERROR: relation "kanji.defenitions" does not exist
japanese_db=> TRUNCATE TABLE kanji.definitions CASCADE;
ERROR: permission denied for table definitions
japanese_db=> TRUNCATE TABLE word.definitions CASCADE;
ERROR: permission denied for table definitions
japanese_db=> TRUNCATE TABLE user_collections.kanji CASCADE;
ERROR: permission denied for table kanji
japanese_db=> SET ROLE privileged_developer
japanese_db-> ;
SET
japanese_db=> TRUNCATE kanji.defenitions CASCADE;
ERROR: relation "kanji.defenitions" does not exist
japanese_db=> TRUNCATE TABLE kanji.defenitions CASCADE;
ERROR: relation "kanji.defenitions" does not exist
japanese_db=> TRUNCATE TABLE kanji.definitions CASCADE;
NOTICE: truncate cascades to table "components"
NOTICE: truncate cascades to table "kanji_map"
NOTICE: truncate cascades to table "kanji_map"
TRUNCATE TABLE
japanese_db=> TRUNCATE TABLE word.definitions CASCADE;
NOTICE: truncate cascades to table "kanji_map"
NOTICE: truncate cascades to table "word_map"
TRUNCATE TABLE
japanese_db=> TRUNCATE TABLE user_collections.kanji CASCADE;
NOTICE: truncate cascades to table "kanji_map"
NOTICE: truncate cascades to table "user_kanji_collection_map"
TRUNCATE TABLE
japanese_db=>
```

Для developer ограничение на выдачу прав:

```
japanese_db=> GRANT ALL PRIVILEGES ON TABLE kanji.definitions TO content_user;
WARNING: no privileges were granted for "definitions"
GRANT
japanese_db=> GRANT ALL PRIVILEGES ON TABLE word.definitions TO content_user;
WARNING: no privileges were granted for "definitions"
GRANT
japanese_db=> GRANT ALL PRIVILEGES ON TABLE user_collections.word TO content_user;
WARNING: no privileges were granted for "word"
GRANT
japanese_db=>
```

Для admin выдача прав и использование TRUNCATE:

```
japanese_db=> GRANT ALL PRIVILEGES ON TABLE kanji.definitions TO content_user;
GRANT
japanese_db=> GRANT ALL PRIVILEGES ON TABLE word.definitions TO content_user;
GRANT
japanese_db=> GRANT ALL PRIVILEGES ON TABLE user_collections.word TO content_user;
GRANT
japanese_db=> TRUNCATE TABLE word.definitions CASCADE;
NOTICE: truncate cascades to table "kanji_map"
NOTICE: truncate cascades to table "word_map"
TRUNCATE TABLE
japanese_db=>
```

## 4. Заключение

В ходе данной работы было выполнено создание схем и ролей для базы данных PostgreSQL.

Для разделения пространства имён и разграничения привелегий было выполнено разделение таблиц на 3 схемы: `kanji`, `word` и `user_collections`. Тем самым названия стали понятнее, а дальнейшая выдача привелегий опиралась не на отдельные таблицы, а на схемы.

Для данной базы данных было предложено создать 5 ролей: `admin`, `developer`, `privileged_developer`, `content_creator`, `content_user`. Каждой роли выделены свои задачи: `admin` — администрирование базы данных, включая создание ролей и выдача привелегий; `developer` — изменение базы данных и схем; `privileged_developer` — роль, для повышения прав `developer`, а именно добавления возможности `TRUNCATE` и создания баз данных; `content_creator` — роль для обогащения словарей иероглифов, слов и системных коллекций; `content_user` — роль для просмотра словарей и создания пользовательских коллекций.

Для проверки привелегий ролей были выведены системные привелегии из таблицы `pg_catalog.pg_roles`. А объектные привелегии были проверены вручную.

## 5. Приложение

Репозиторий: GitHub