

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНОМУ УНІВЕРСИТЕТІ “ЛЬВІВСЬКА
ПОЛІТЕХНІКА”**



Розрахункова робота
на тему «Помічник пошуку добавок»
з дисципліни «Організація баз даних та знань»

Виконав:
студент групи КН-209
Мацкула Олег
Викладач:
Мельникова Н. І.

Львів – 2020 р.

ЗМІСТ

1. Тема проекту.....	3
2. Вступ.....	4
3. Логічна схема БД	7
4. Опис структури БД.	8
5. Фізична модель БД.	10
6. Ділова модель.	14
7. Запити до БД.	15
8. Висновки.	19
9. Список літератури.....	20

1. Тема проекту

Наш проект має за мету зібрати в одному місці всю інформацію про харчові добавки, в єдиному форматі та структурі. Також, ми хочемо надати всім бажаючим простий та швидкий доступ до неї.

Обговоривши ідею ми вирішили створити два інтерфеси доступу та власний ППІ. Для цього ми використати SQL базу даних PostgreSQL.

Ознайомитись із реалізацією можна за такими посиланнями:

Telegram бот - t.me/project_healthy_bot

Веб-аплакація - 78.46.244.156

2. Вступ

Щодня ми задумуємось про органічність та склад продуктів, які ми вживаємо. Але де ж взяти таку інформацію? В інтернеті? 80% інформації в мережі – фейки. Знайти у декількох джерелах схожу інформацію? Хто ж буде тратити на це час... Отож, ми вирішили створити проект Healthy. Тепер кожен, хто має доступ до інтернету зможе скористатись одним із наших інтерфейсів для пошуку інформації про ту чи іншу харову добавку.

Хорошим стартом було придумати логічну схему нашої майбутньої бази даних, після цього важливого кроку слід було розробити фізичну модель, а на останок залишилось всього лиш поступово заповнювати даними.

Визначення та опис предметної області

Назва: "Довідник харчових добавок"

Основні сутності: "Добавка", "Клас", "Призначення", "Побічні ефекти"

Інші сутності: "Корстувач", "Сесія" і тд.

Найчастіші запити: Запит пошуку добавки по імені та Е-коду; запит вибору всіх добавок з певним класом, призначенням, або побічним ефектом; запит вибору всіх побічних ефектів певної добавки.

Модель повинна містити у собі усі основні сутності. Вона повинна надати доступ до усіх Добавок. Кожна добавка має свій клас та певне призначення, а також декілька побічних ефектів. Різні добавки можуть мати однакові побічні ефекти, класи та призначення.

Наші переваги

1. Компактність

Ми додаємо у наш проєкт лише потрібний мінімум функціоналу, до того ж людям не потрібно завантажувати нічого зайвого на свій девайс

2. Доступ до інтернету – єдина потрібна річ для користування сервісом

3. Приємний інтерфейс

Для того щоб користувачі отримували лише позитивні емоції, ми створили гарний та зрозумілий інтерфейс.

4. Доцільність

На даний момент багато людей хвилюються за якість і натуральність їжі і засобів догляду, якими вони користуються, і усе більше й більше людей намагаються контролювати органічність усіх продуктів. Тому наш проект достатньо активно буде використовуватись усіма людьми, які хвилюються за своє здоров'я і навколишнє середовище.

5. Можливість отримувати заохочуючі внески від аудиторії

Так як багато зрілих і дорослих людей буде користуватись нашою аплікацією, вони зможуть вносити платежі наприклад, для отримання преміум контенту або для користування нашим ППІ.

6. Відсутність конкуренції

На ринку майже немає сервісів для отримання інформації по складу, а ті що ми знайшли – "киво" зроблені.

7. Простота доступу

Для ще простішого доступу до нашого сервісу, ми створили Telegram бота.

8. Правдивість інформації від надійних джерел

Ми переконуємось у достовірності інформації лише тоді, коли знаходимо однакові дані у декількох місцях. За схожим правилом визначають правдивість історичних подій. Тому наші користувачі не повинні сумніватись у нашій чесності.

Цілісність даних

Обмеження цілісності запобігають появі в БД суперечливих даних. Вирішення цієї проблеми полягає у такому:

- наявність обов'язкових і необов'язкових значень даних для атрибутів (NULL, NOT NULL);

У нашій базі наявні деякі необов'язкові поля, які не сильно важливі у даній моделі і їхня відсутність не є великою проблемою.

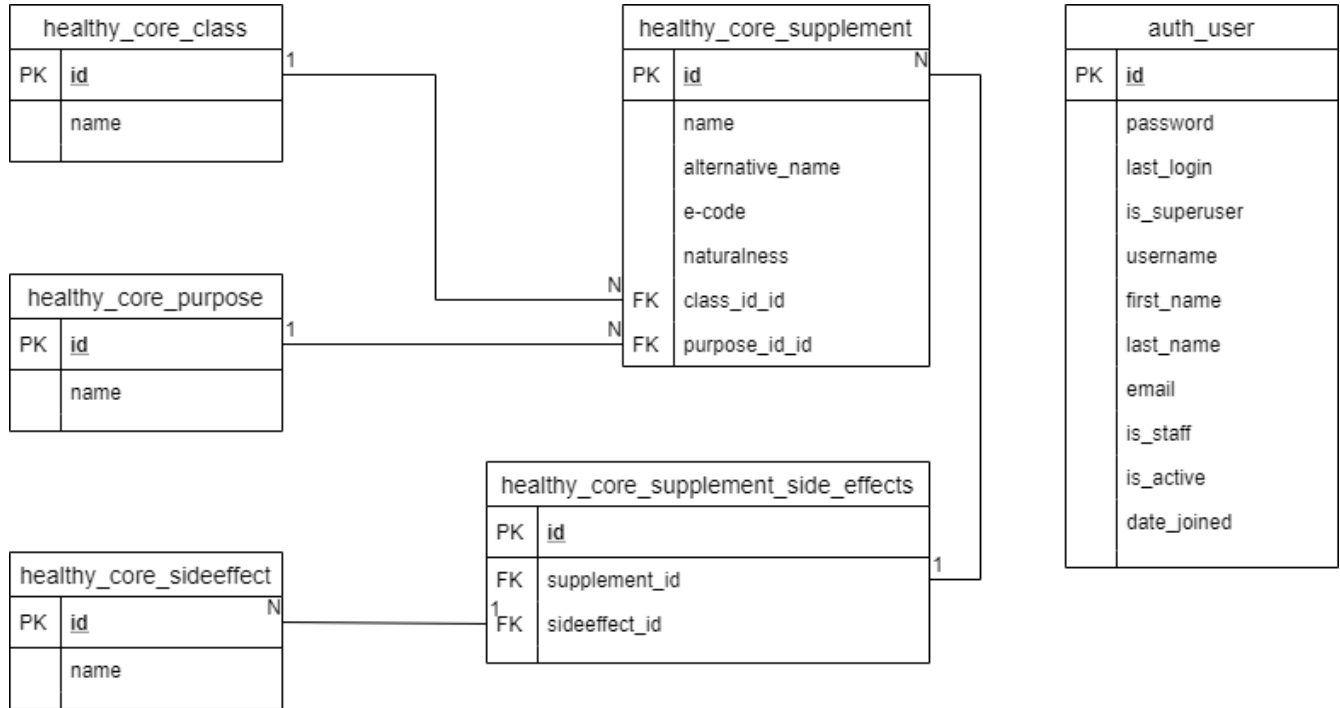
- цілісність сутностей (обов'язкова наявність Primary Key в кожному відношенні);

У нашій базі даних у кожної сутності наявний обов'язковий Primary Key для цілісності.

- посилкова цілісність (зв'язування таблиць за допомогою Foreign Key);

Також щоб забезпечити цілісність зв'язків між сутностями наявний зовнішній ключ. Для безперешкодного отримування інформації між даними таблицям

3. Логічна схема



4. Опис структури

Таблиця Класів(class):

- id serial – первинний ключ, цілочисельний унікальний ідентифікатор.
- name varchar(128) – назва класу добавки.

Таблиця Призначення(purpose):

- id serial – первинний ключ, цілочисельний унікальний ідентифікатор.
- name varchar(128) – назва призначення добавки

Таблиця Побічні ефекти(side_effect):

- id serial – первинний ключ, цілочисельний унікальний ідентифікатор.
- name varchar(128) – назва побічних ефектів добавок

Таблиця проміжна між побічними ефектами та добавкою
(supplement_side_effect):

- id serial – первинний ключ, цілочисельний унікальний ідентифікатор.
- side_effect_id int – зовнішній ключ до сутності побічних ефектів
- supplement_id int – зовнішній ключ до сутності добавок

Таблиця Добавки(supplement):

- id serial – первинний ключ, цілочисельний унікальний ідентифікатор.
- name varchar(128)– назва побічних ефектів добавок
- alt_name varchar(128)– додаткова назва добавки
- e_code varchar(16)– е-номер добавки
- naturalness boolean – натуральність добавки
- class_id int – зовнішній ключ до сутності класів.(один – до батьох)
- purpose_id int – зовнішній ключ до сутності призначення(один – до багатьох)

- side_effects_id int – зв’язок зі сутністю Побічних ефектів багато – до багатьох.

Таблиця Користувача(user):

- id serial – первинний ключ, цілочисельний унікальний ідентифікатор.
- password varchar(128) – пароль для профілю користувача
- last_login timestamp with time zone – дата останнього входу на дану сторінку
- is_superuser boolean – ідентифікатор адміністраторів сайту
- username varchar(150) – ідентифікатор людини на нашому сайті
- first_name varchar(30) – ім’я користувача
- last_name varchar(150) – прізвище користувача
- email varchar(254) – електронна пошта користувача
- is_staff boolean – ідентифікатор працівника сайту
- is_active boolean – ідентифікатор активності користувача протягом останнього

місяця

- date_joined timestamp with time zone – дата реєстрації на сайті

5. Фізична модель

```
create table healthy_core_purpose
(
    id      serial          not null
        constraint healthy_core_purpose_pkey
            primary key,
    name varchar(128) not null
);
create table healthy_core_class
(
    id      serial          not null
        constraint healthy_core_class_pkey
            primary key,
    name varchar(128) not null
);
create table healthy_core_sideeffect
(
    id      serial          not null
        constraint healthy_core_sideeffect_pkey
            primary key,
    name varchar(128) not null
);
create table healthy_core_supplement
(
    id      serial          not null
        constraint healthy_core_supplement_pkey
            primary key,
    name      varchar(128) not null,
    alternative_name varchar(128) not null,
    e_code     varchar(16)  not null,
    naturalness boolean      not null,
```

```

        class_id_id        integer        not null
            constraint
healthy_core_supplem_class_id_id_8de11e97_fk_healthy_c
            references healthy_core_class

        purpose_id_id      integer        not null
            constraint
healthy_core_supplem_purpose_id_id_e4b3a38c_fk_healthy_c
            references healthy_core_purpose

);
create table healthy_core_supplement_side_effects
(
    id serial not null
        constraint healthy_core_supplement_side_effects_pkey
        primary key,
    supplement_id integer not null
        constraint
healthy_core_supplem_supplement_id_4586e3a0_fk_healthy_c
        references healthy_core_supplement
    sideeffect_id integer not null
        constraint
healthy_core_supplem_sideeffect_id_6cd096fb_fk_healthy_c
        references healthy_core_sideeffect
    constraint
healthy_core_supplement__supplement_id_sideeffect_c307f911_
    uniq
        unique (supplement_id, sideeffect_id)
);

create table auth_user

```

```

(
    id            serial            not null
        constraint auth_user_pkey
            primary key,
    password      varchar(128)       not null,
    last_login    timestamp with time zone,
    is_superuser  boolean            not null,
    username      varchar(150)       not null
        constraint auth_user_username_key
            unique,
    first_name    varchar(30)        not null,
    last_name     varchar(150)       not null,
    email         varchar(254)       not null,
    is_staff      boolean            not null,
    is_active     boolean            not null,
    date_joined   timestamp with time zone not null
);

create index auth_user_username_6821ab7c_like
on auth_user (username);

create unique index healthy_core_class_name_uindex
on healthy_core_class (name);

create unique index healthy_core_purpose_name_uindex
on healthy_core_purpose (name);

create unique index healthy_core_sideeffect_name_uindex
on healthy_core_sideeffect (name);

create index healthy_core_supplement_class_id_id_8de11e97
on healthy_core_supplement (class_id_id);

create index healthy_core_supplement_purpose_id_id_e4b3a38c
on healthy_core_supplement (purpose_id_id);

```

```
create index
healthy_core_supplement_side_effects_supplement_id_4586e3a0
  on healthy_core_supplement_side_effects (supplement_id);
```

```
create index
healthy_core_supplement_side_effects_sideeffect_id_6cd096fb
  on healthy_core_supplement_side_effects (sideeffect_id);
```

6. Ділова модель

Ділова модель бази даних визначена нижче. Вона встановлює відповідність між певною функцією, яку виконує сервіс, та сутністю в базі даних, яка бере участь в запиті. Усі запити, які я показав будуть відображені тут.

Функції	Класи	Призначення	Побічні ефекти	Добавки	Користувач
Кількість добавок				*	
Пошук користувача за датою реєстрації					*
Групування за класами, і підрахунок	*			*	
Отримання добавок за їх призначенням		*		*	
Отримання добавок за побічним ефектом			*	*	

7. Запити

Кількість добавок в базі

Запит:

```
SELECT count(*)  
FROM healthy_core_supplement
```

Відповідь:

	count
1	65

Пошук користувача за датою реєстрації

Запит:

```
SELECT username, date_joined  
FROM auth_user  
WHERE date_joined >= '2020-05-01' AND date_joined < '2020-06-01';
```

Відповідь:

	username	date_joined
1	admin	2020-05-22 20:21:54.580793

Групування за класами і підрахунок

Запит:

```
SELECT c.name class_name, count(*)  
FROM healthy_core_supplement s INNER JOIN healthy_core_class c on  
s.class_id_id = c.id  
GROUP BY c.name;
```

Відповідь:

	class_name	count
1	Neutral	38
2	Healthy	14
3	Unsafe	6
4	Harmful	7

Отримання добавок за їх призначенням

Код функції:

```
CREATE OR REPLACE FUNCTION  
supplements_with_purpose(purpose_name varchar)  
RETURNS TABLE (  
    supplement_id int,  
    supplement_name varchar  
) AS $$  
BEGIN  
    RETURN QUERY SELECT s.id, s.name  
    FROM healthy_core_supplement s INNER JOIN  
healthy_core_purpose p on s.purpose_id_id = p.id  
    WHERE p.name = purpose_name;  
END $$  
LANGUAGE 'plpgsql';
```


Запит:

```
SELECT *  
FROM supplements_with_purpose('Colours');
```

Відповідь:

supplement_id	supplement_name
1	2 Curcumin
2	20 Riboflavin
3	21 Saffron
4	33 Tartrazine
5	34 Alkannin
6	35 Carmoisine
7	36 Carbon black
8	37 Lutein
9	38 Lycopene
10	64 Azorubine
11	65 Amaranth
12	66 Erythrosine
13	84 Allura Red AC

Отримання добавок за побічним ефектом

Код функції

```
CREATE OR REPLACE FUNCTION  
supplements_with_side_effect(se_name varchar)  
RETURNS TABLE (  
    supplement_id int,  
    supplement_name varchar  
) AS $$  
BEGIN  
    RETURN QUERY SELECT s.id, s.name  
    FROM healthy_core_supplement s  
        INNER JOIN healthy_core_supplement_side_effects sse on s.id  
= sse.supplement_id  
        INNER JOIN healthy_core_sideeffect se on sse.sideeffect_id =  
se.id
```

```
WHERE se.name = se_name;  
END $$  
LANGUAGE 'plpgsql';
```

Запит

```
SELECT *  
FROM supplements_with_side_effect('Skin rash');
```

Відповідь

	supplement_id	supplement_name
1	10	Bacitracin
2	13	Cholic acid
3	19	Glutamic acid
4	34	Alkannin
5	35	Carmoisine
6	39	Potassium sorbate
7	64	Azorubine
8	83	Tocopherols
9	85	Propyl Gallate

8. Висновки

Виконуючи дану розрахункову роботу я навчився проектувати моделі баз даних для предметних областей. Реалізував фізичну модель та показав декілька запитів які використовуються при роботі сервісу. Усі запити на сервер PostgreSQL логуються і доступні для перегляду. Дізнався багато нового про бази даних та їх адміністрування загалом.

9. Використані джерела

- [1] <https://www.postgresql.org/download/>
- [2] <https://www.postgresql.org/docs/>
- [3] <https://en.wikipedia.org/wiki/Database>
- [4] <https://www.oracle.com/database/what-is-database.html>
- [5] <https://www.mysql.com>
- [6] <https://www.coursera.org/learn/sql-data-science>
- [7] <https://www.postgresql.org/docs/9.1/sql-createfunction.html>
- [8] <https://www.coursera.org/learn/sql-for-data-science>
- [9] [https://uk.wikipedia.org/wiki/Нормалізація баз даних](https://uk.wikipedia.org/wiki/Нормалізація_баз_даних)
- [10] <https://www.studytonight.com/dbms/database-normalization.php>
- [11] <https://www.guru99.com/database-normalization.html>