База данных онлайн-сервиса цифрового распространения компьютерных игр и программ «Steam».

В этом проекте я хотел реализовать основу для интернета магазина/сервиса по продажам игр и программ.

В этой базе данных пользователь сможет:

1) Зарегистрироваться(users), заполнить данные своего профиля(profiles)

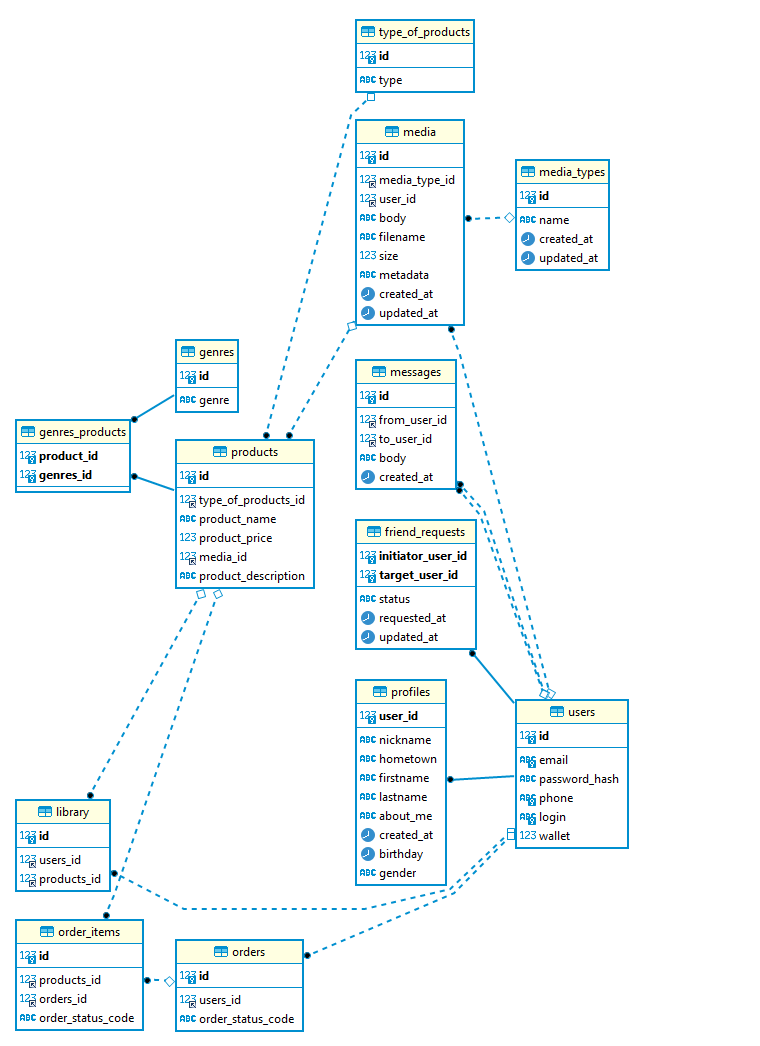
2) Вести светскую беседу на тему интернет технологий(messages)

3) Добавлять людей в друзья(friends\_requests)

4) Совершать выгодные покупки на великолепной площадке(orders)

5) Просматривать большой выбор игр и программ на любой вкус, цвет и жанр(products)

6) Так же продукты разделены на свои жанры(genres)(genres\_products) и типы (type\_of\_products)



Рассмотрим подробнее каждую таблицу.

Таблица **users**.

Одна из ключевых таблиц нашей базы данных это таблица **users**.

* ID – первичный ключ, bigint unsigned т.к. пользователей планируется много, not null, auto increment для авто увеличения числа.
* Email – varchar (150), not null, и является unique. Тут хранится почта юзера.
* Password\_hash – varchar (100), not null, и является unique. Здесь находится хэш пароля пользователя.
* Phone – varchar (100), not null. Соответственно телефон пользователя.
* Login – varchar (200), not null. Логин пользователя
* Wallet – float, not null. Кошелек пользователя для оплаты покупок.

(можно каждую таблицу расписывать подобным способом)

Таблица **profiles.**

Находится основная информация нашего пользователя.

Первичным ключом является users\_id и ссылается на id таблицу **users.** Связь реализована 1 к 1, 1 профиль к 1 юзеру.

Таблица **friends\_requests.**

Таблица дружбы между юзерами.

Тут реализован составной первичный ключ, который состоит из двух столбцов initiator\_user\_id и target\_user\_id оба ссылаются на id таблицу **users**.

Status имеет тип данных enum('requested','approved','declined','unfriended').

Таблица **messages**.

Первичный ключ ID, так же присутствуют initiator\_user\_id и target\_user\_id.

Body имеет тип данных text для хранения текста сообщения.

Таблица **media.**

Таблица необходимая для хранения медиа файлов. Имеет первичный ключ ID.

**Sql Schema.**

CREATE TABLE `friend\_requests` (

`initiator\_user\_id` bigint unsigned NOT NULL,

`target\_user\_id` bigint unsigned NOT NULL,

`status` enum('requested','approved','declined','unfriended') CHARACTER SET utf8mb4 COLLATE utf8mb4\_unicode\_ci DEFAULT NULL,

`requested\_at` datetime DEFAULT CURRENT\_TIMESTAMP,

`updated\_at` datetime DEFAULT NULL ON UPDATE CURRENT\_TIMESTAMP,

PRIMARY KEY (`initiator\_user\_id`,`target\_user\_id`),

CONSTRAINT `friend\_requests\_FK` FOREIGN KEY (`initiator\_user\_id`) REFERENCES `users` (`id`) );

CREATE TABLE `genres` (

`id` bigint unsigned NOT NULL AUTO\_INCREMENT,

`genre` varchar(100) NOT NULL,

PRIMARY KEY (`id`));

CREATE TABLE `genres\_products` (

`product\_id` bigint unsigned NOT NULL,

`genres\_id` bigint unsigned NOT NULL,

PRIMARY KEY (`product\_id`,`genres\_id`),

KEY `genres\_products\_FK` (`genres\_id`),

CONSTRAINT `genres\_products\_FK` FOREIGN KEY (`genres\_id`) REFERENCES `genres` (`id`),

CONSTRAINT `genres\_products\_FK\_1` FOREIGN KEY (`product\_id`) REFERENCES `products` (`id`));

CREATE TABLE `library` (

`id` bigint unsigned NOT NULL AUTO\_INCREMENT,

`users\_id` bigint unsigned NOT NULL,

`products\_id` bigint unsigned NOT NULL,

PRIMARY KEY (`id`),

KEY `library\_FK` (`users\_id`),

KEY `library\_FK\_1` (`products\_id`),

CONSTRAINT `library\_FK` FOREIGN KEY (`users\_id`) REFERENCES `users` (`id`),

CONSTRAINT `library\_FK\_1` FOREIGN KEY (`products\_id`) REFERENCES `products` (`id`));

CREATE TABLE `media` (

`id` bigint unsigned NOT NULL AUTO\_INCREMENT,

`media\_type\_id` bigint unsigned NOT NULL,

`user\_id` bigint unsigned NOT NULL,

`body` text CHARACTER SET utf8mb4 COLLATE utf8mb4\_unicode\_ci,

`filename` varchar(255) CHARACTER SET utf8mb4 COLLATE utf8mb4\_unicode\_ci DEFAULT NULL,

`size` int DEFAULT NULL,

`metadata` longtext CHARACTER SET utf8mb4 COLLATE utf8mb4\_bin,

`created\_at` datetime DEFAULT CURRENT\_TIMESTAMP,

`updated\_at` datetime DEFAULT NULL ON UPDATE CURRENT\_TIMESTAMP,

UNIQUE KEY `id` (`id`),

KEY `media\_type\_id` (`media\_type\_id`) USING BTREE,

KEY `user\_id` (`user\_id`) USING BTREE,

CONSTRAINT `media\_FK` FOREIGN KEY (`media\_type\_id`) REFERENCES `media\_types` (`id`),

CONSTRAINT `media\_FK\_1` FOREIGN KEY (`user\_id`) REFERENCES `users` (`id`));

CREATE TABLE `media\_types` (

`id` bigint unsigned NOT NULL AUTO\_INCREMENT,

`name` varchar(255) CHARACTER SET utf8mb4 COLLATE utf8mb4\_unicode\_ci DEFAULT NULL,

`created\_at` datetime DEFAULT CURRENT\_TIMESTAMP,

`updated\_at` datetime DEFAULT NULL ON UPDATE CURRENT\_TIMESTAMP,

PRIMARY KEY (`id`));

CREATE TABLE `messages` (

`id` bigint unsigned NOT NULL AUTO\_INCREMENT,

`from\_user\_id` bigint unsigned NOT NULL,

`to\_user\_id` bigint unsigned NOT NULL,

`body` text CHARACTER SET utf8mb4 COLLATE utf8mb4\_unicode\_ci,

`created\_at` datetime DEFAULT CURRENT\_TIMESTAMP,

UNIQUE KEY `id` (`id`),

KEY `from\_user\_id` (`from\_user\_id`) USING BTREE,

KEY `to\_user\_id` (`to\_user\_id`) USING BTREE,

CONSTRAINT `messages\_FK` FOREIGN KEY (`from\_user\_id`) REFERENCES `users` (`id`),

CONSTRAINT `messages\_FK\_1` FOREIGN KEY (`to\_user\_id`) REFERENCES `users` (`id`));

CREATE TABLE `order\_items` (

`id` bigint unsigned NOT NULL AUTO\_INCREMENT,

`products\_id` bigint unsigned NOT NULL,

`orders\_id` bigint unsigned NOT NULL,

`order\_status\_code` enum('Cancelled','Completed') CHARACTER SET utf8mb4 COLLATE utf8mb4\_unicode\_ci NOT NULL,

PRIMARY KEY (`id`),

KEY `order\_items\_FK` (`orders\_id`),

KEY `order\_items\_FK\_1` (`products\_id`),

CONSTRAINT `order\_items\_FK` FOREIGN KEY (`orders\_id`) REFERENCES `orders` (`id`),

CONSTRAINT `order\_items\_FK\_1` FOREIGN KEY (`products\_id`) REFERENCES `products` (`id`));

CREATE TABLE `orders` (

`id` bigint unsigned NOT NULL AUTO\_INCREMENT,

`users\_id` bigint unsigned NOT NULL,

`order\_status\_code` enum('Cancelled','Completed') CHARACTER SET utf8mb4 COLLATE utf8mb4\_unicode\_ci NOT NULL,

PRIMARY KEY (`id`),

KEY `orders\_FK` (`users\_id`),

CONSTRAINT `orders\_FK` FOREIGN KEY (`users\_id`) REFERENCES `users` (`id`));

CREATE TABLE `products` (

`id` bigint unsigned NOT NULL AUTO\_INCREMENT,

`type\_of\_products\_id` bigint unsigned NOT NULL,

`product\_name` varchar(200) NOT NULL,

`product\_price` float NOT NULL,

`media\_id` bigint unsigned NOT NULL,

`product\_description` text NOT NULL,

PRIMARY KEY (`id`),

KEY `products\_FK` (`media\_id`),

KEY `products\_FK\_1` (`type\_of\_products\_id`),

CONSTRAINT `products\_FK` FOREIGN KEY (`media\_id`) REFERENCES `media` (`id`),

CONSTRAINT `products\_FK\_1` FOREIGN KEY (`type\_of\_products\_id`) REFERENCES `type\_of\_products` (`id`));

CREATE TABLE `profiles` (

`user\_id` bigint unsigned NOT NULL,

`nickname` varchar(100) NOT NULL,

`hometown` varchar(100) DEFAULT NULL,

`firstname` varchar(100) DEFAULT NULL,

`lastname` varchar(100) DEFAULT NULL,

`about\_me` varchar(250) DEFAULT NULL,

`created\_at` datetime NOT NULL DEFAULT CURRENT\_TIMESTAMP,

`birthday` date DEFAULT NULL,

`gender` char(1) CHARACTER SET utf8mb4 COLLATE utf8mb4\_unicode\_ci DEFAULT NULL,

UNIQUE KEY `profiles\_UN` (`user\_id`),

CONSTRAINT `profiles\_FK` FOREIGN KEY (`user\_id`) REFERENCES `users` (`id`) ON DELETE CASCADE ON UPDATE CASCADE);

CREATE TABLE `type\_of\_products` (

`id` bigint unsigned NOT NULL AUTO\_INCREMENT,

`type` varchar(100) NOT NULL,

PRIMARY KEY (`id`));

CREATE TABLE `users` (

`id` bigint unsigned NOT NULL AUTO\_INCREMENT,

`email` varchar(150) NOT NULL,

`password\_hash` varchar(100) NOT NULL,

`phone` varchar(100) NOT NULL,

`login` varchar(200) NOT NULL,

`wallet` float NOT NULL,

PRIMARY KEY (`id`),

UNIQUE KEY `phone` (`phone`),

UNIQUE KEY `email` (`email`),

UNIQUE KEY `login` (`login`));