

**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ
ФЕДЕРАЦИИ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ “ВЫСШАЯ ШКОЛА
ЭКОНОМИКИ”**

Факультет компьютерных наук
Департамент Программной инженерии

Пояснительная записка
Для командного проекта по курсу “Базы данных”

Выполнили:

Лоозе Тимофей БПИ205

Колб Илья БПИ205

Шишлянников Александр БПИ205

Проверил:

Брейман А. Д.

Москва 2022

Сфера применения.	3
Функциональные требования.	4
Нефункциональные требования.	6
UML-диаграмма.	7
Функциональные зависимости.	8
DDL.	9
SQL-запросы.	13

Сфера применения.

Проект представляет из себя внутренний сервис для маркетплейсов. Благодаря данному сервису поставщики и продавцы товаров смогут контактировать, заключать контракты, проводить аукционы купли-продажи товаров.

Проект является уникальным и достаточно универсальным, его можно использовать в любом маркетплейсе.

Пользователи данного сервиса – продавцы и поставщики товаров. Сервис администрируется наделенными данным правом пользователями (внутренними сотрудниками сервиса).

Функциональные требования.

Требования относительно пользователей.

Пользователь должен иметь возможность:

1. Зарегистрироваться в системе под одной из категорий:
 - a. Продавец
 - b. Поставщик
 - c. Администратор
2. Зарегистрироваться в системе под категорией “продавец”, введя следующие данные:
 - a. Электронная почта
 - b. Номер телефона
 - c. Пароль
 - d. ФИО
 - e. ИНН
 - f. Регион
3. Зарегистрироваться в системе под категорией “поставщик”, введя следующие данные:
 - a. Электронная почта
 - b. Номер телефона
 - c. Пароль
 - d. ФИО
 - e. ИНН
 - f. Регион
4. Указать в системе, какой тип товара он поставляет/продает, если пользователь является поставщиком или продавцом соответственно.

Требования относительно личного кабинета пользователя.

Пользователь должен иметь возможность:

1. Изменить или добавить дополнительную информацию в личном кабинете.
2. Удалить аккаунт.

Требования относительно купли-продажи товара.

Поставщик должен иметь возможность:

1. Разместить товар на продажу с указанием следующей информации о товаре:
 - a. Название
 - b. Бренд
 - c. Количество
 - d. Цена за штуку
 - e. Объём
 - f. Комиссия
2. Заключить с продавцом контракт на поставку с указанием типа оплаты и типа контракта.

3. По соглашению с продавцом редактировать контракт.
4. Отозвать контракт с продавцом

Продавец должен иметь возможность:

1. Заключить с поставщиком контракт на поставку с указанием типа оплаты и типа контракта.
2. По соглашению с поставщиком редактировать контракт.
3. Отозвать контракт с поставщиком.

Требования относительно товаров.

Поставщики и продавцы должны иметь возможность:

1. Указывать в системе местонахождение товара.

Требования относительно возможностей администратора.

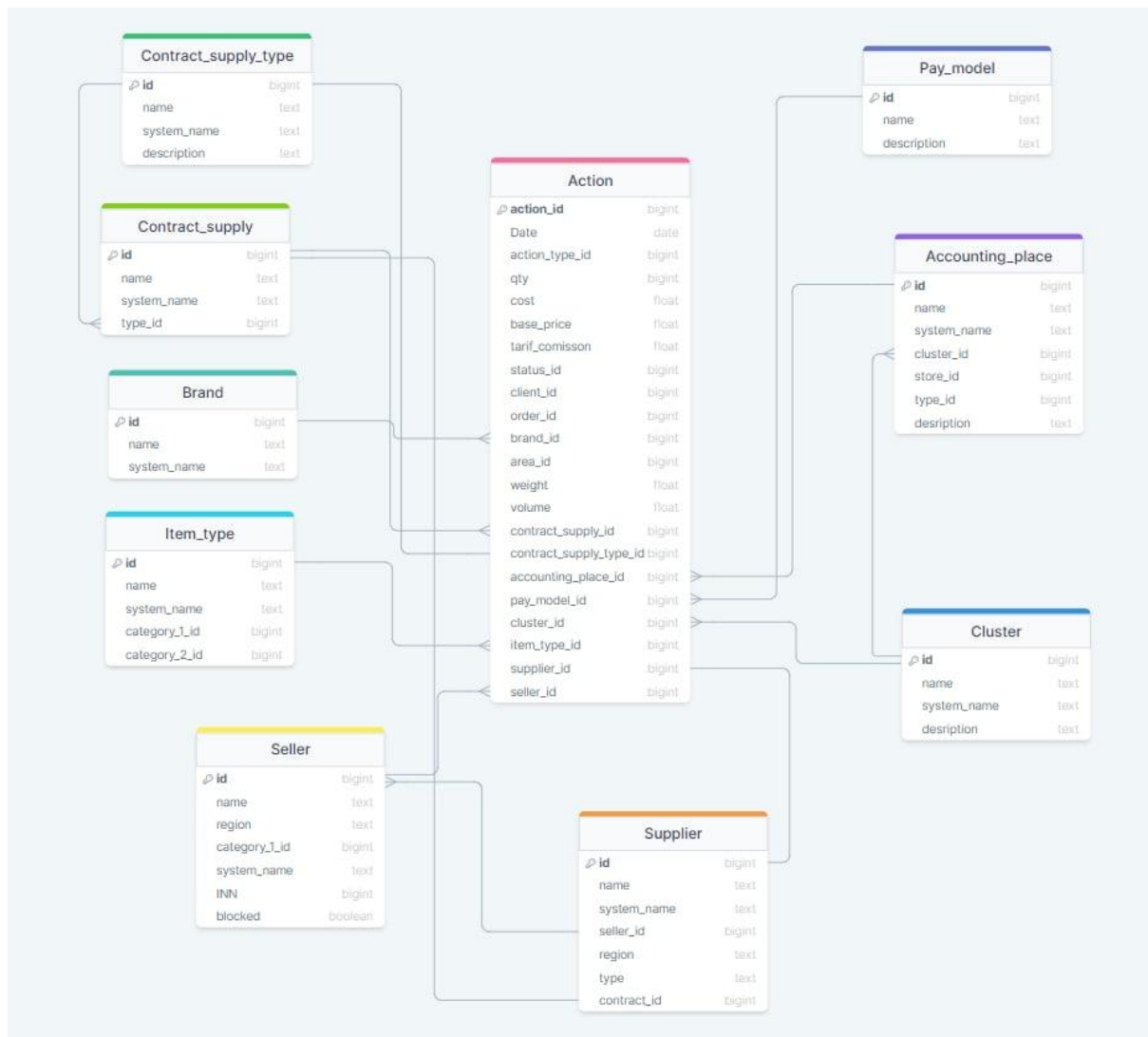
Администратор должен иметь возможность:

1. Входить в систему с данными ему почтой и паролем.
2. Просматривать все заключенные контракты.
3. Аннулировать незаконные контракты.
4. Просматривать информацию о всех зарегистрированных в системе поставщиках и продавцах.

Нефункциональные требования.

1. Сервис должен быть доступным в браузерах Google Chrome, Yandex, Safari, Firefox, Opera, Microsoft Edge.
2. Сервис должен быть доступен 99.9% времени.
3. Сервис должен иметь интуитивно понятный интерфейс.
4. Ответы от сервера должны приходить не позже, чем через 30 секунд.
5. Сервис должен обеспечивать защищенность введенных пользователями данных.

UML-диаграмма.



Функциональные зависимости.

Item_type:

{id} -> {name, system_name, category_1_id, category_2_id}

Seller:

{id} -> {name, system_name, category_1_id, region, INN, blocked}

Supplier:

{id} -> {name, system_name, seller_id, region, type, contact_id}

Contract_supply:

{id} -> {name, system_name, type_id}

Action:

{action_id} -> {Date, action_type_id, qly, cost, base_price, tariff_comission, status_id, client_id, order_id, brand_id, area_id, weight, volume, contract_supply_id, contract_supply_type_id, accounting_price_id, pay_model_id, cluster_id, item_type_id, supplier_id, seller_id}

Accounting_place:

{id} -> {name, system_name, cluster_id, store_id, type_id, description}

Cluster:

{id} -> {name, system_name, description}

Contract_supply_type:

{id} -> {name, system_name, description}

Brand:

{id} -> {name, system_name}

Pay_model:

{id} -> {name, description}

DDL.

```
CREATE TABLE "Action"(  
  "action_id" BIGINT NOT NULL,  
  "Date" DATE NOT NULL,  
  "action_type_id" BIGINT NOT NULL,  
  "qty" BIGINT NOT NULL,  
  "cost" DOUBLE PRECISION NOT NULL,  
  "base_price" DOUBLE PRECISION NOT NULL,  
  "tarif_comisson" DOUBLE PRECISION NOT NULL,  
  "status_id" BIGINT NOT NULL,  
  "client_id" BIGINT NOT NULL,  
  "order_id" BIGINT NOT NULL,  
  "brand_id" BIGINT NOT NULL,  
  "area_id" BIGINT NOT NULL,  
  "weight" DOUBLE PRECISION NOT NULL,  
  "volume" DOUBLE PRECISION NOT NULL,  
  "contract_supply_id" BIGINT NOT NULL,  
  "contract_supply_type_id" BIGINT NOT NULL,  
  "accounting_place_id" BIGINT NOT NULL,  
  "pay_model_id" BIGINT NOT NULL,  
  "cluster_id" BIGINT NOT NULL,  
  "item_type_id" BIGINT NOT NULL,  
  "supplier_id" BIGINT NOT NULL,  
  "seller_id" BIGINT NOT NULL  
);  
ALTER TABLE  
  "Action" ADD PRIMARY KEY("action_id");  
CREATE TABLE "Accounting_place"(  
  "id" BIGINT NOT NULL,  
  "name" TEXT NOT NULL,  
  "system_name" TEXT NOT NULL,  
  "cluster_id" BIGINT NOT NULL,  
  "store_id" BIGINT NOT NULL,  
  "type_id" BIGINT NOT NULL,  
  "description" TEXT NOT NULL  
);  
ALTER TABLE  
  "Accounting_place" ADD PRIMARY KEY("id");  
CREATE TABLE "Pay_model"(  
  "id" BIGINT NOT NULL,  
  "name" TEXT NOT NULL,  
  "description" TEXT NOT NULL
```

```

);
ALTER TABLE
    "Pay_model" ADD PRIMARY KEY("id");
CREATE TABLE "Cluster"(
    "id" BIGINT NOT NULL,
    "name" TEXT NOT NULL,
    "system_name" TEXT NOT NULL,
    "desription" TEXT NOT NULL
);
ALTER TABLE
    "Cluster" ADD PRIMARY KEY("id");
CREATE TABLE "Brand"(
    "id" BIGINT NOT NULL,
    "name" TEXT NOT NULL,
    "system_name" TEXT NOT NULL
);
ALTER TABLE
    "Brand" ADD PRIMARY KEY("id");
CREATE TABLE "Item_type"(
    "id" BIGINT NOT NULL,
    "name" TEXT NOT NULL,
    "system_name" TEXT NOT NULL,
    "category_1_id" BIGINT NOT NULL,
    "category_2_id" BIGINT NOT NULL
);
ALTER TABLE
    "Item_type" ADD PRIMARY KEY("id");
CREATE TABLE "Contract_supply"(
    "id" BIGINT NOT NULL,
    "name" TEXT NOT NULL,
    "system_name" TEXT NOT NULL,
    "type_id" BIGINT NOT NULL
);
ALTER TABLE
    "Contract_supply" ADD PRIMARY KEY("id");
CREATE TABLE "Contract_supply_type"(
    "id" BIGINT NOT NULL,
    "name" TEXT NOT NULL,
    "system_name" TEXT NOT NULL,
    "description" TEXT NOT NULL
);
ALTER TABLE
    "Contract_supply_type" ADD PRIMARY KEY("id");
CREATE TABLE "Seller"(

```

```

    "id" BIGINT NOT NULL,
    "name" TEXT NOT NULL,
    "region" TEXT NOT NULL,
    "category_1_id" BIGINT NOT NULL,
    "system_name" TEXT NOT NULL,
    "INN" BIGINT NOT NULL,
    "blocked" BOOLEAN NOT NULL
);
ALTER TABLE
    "Seller" ADD PRIMARY KEY("id");
CREATE TABLE "Supplier"(
    "id" BIGINT NOT NULL,
    "name" TEXT NOT NULL,
    "system_name" TEXT NOT NULL,
    "seller_id" BIGINT NOT NULL,
    "region" TEXT NOT NULL,
    "type" TEXT NOT NULL,
    "contract_id" BIGINT NOT NULL
);
ALTER TABLE
    "Supplier" ADD PRIMARY KEY("id");
ALTER TABLE
    "Action" ADD CONSTRAINT "action_accounting_place_id_foreign" FOREIGN
KEY("accounting_place_id") REFERENCES "Accounting_place"("id");
ALTER TABLE
    "Action" ADD CONSTRAINT "action_supplier_id_foreign" FOREIGN
KEY("supplier_id") REFERENCES "Supplier"("id");
ALTER TABLE
    "Action" ADD CONSTRAINT "action_seller_id_foreign" FOREIGN KEY("seller_id")
REFERENCES "Seller"("id");
ALTER TABLE
    "Action" ADD CONSTRAINT "action_pay_model_id_foreign" FOREIGN
KEY("pay_model_id") REFERENCES "Pay_model"("id");
ALTER TABLE
    "Action" ADD CONSTRAINT "action_cluster_id_foreign" FOREIGN KEY("cluster_id")
REFERENCES "Cluster"("id");
ALTER TABLE
    "Accounting_place" ADD CONSTRAINT "accounting_place_cluster_id_foreign"
FOREIGN KEY("cluster_id") REFERENCES "Cluster"("id");
ALTER TABLE
    "Supplier" ADD CONSTRAINT "supplier_seller_id_foreign" FOREIGN
KEY("seller_id") REFERENCES "Seller"("id");
ALTER TABLE

```

```
"Supplier" ADD CONSTRAINT "supplier_contract_id_foreign" FOREIGN  
KEY("contract_id") REFERENCES "Contract_supply"("id");
```

```
ALTER TABLE
```

```
"Action" ADD CONSTRAINT "action_contract_supply_type_id_foreign" FOREIGN  
KEY("contract_supply_type_id") REFERENCES "Contract_supply_type"("id");
```

```
ALTER TABLE
```

```
"Contract_supply" ADD CONSTRAINT "contract_supply_type_id_foreign" FOREIGN  
KEY("type_id") REFERENCES "Contract_supply_type"("id");
```

```
ALTER TABLE
```

```
"Action" ADD CONSTRAINT "action_contract_supply_id_foreign" FOREIGN  
KEY("contract_supply_id") REFERENCES "Contract_supply"("id");
```

```
ALTER TABLE
```

```
"Action" ADD CONSTRAINT "action_brand_id_foreign" FOREIGN KEY("brand_id")  
REFERENCES "Brand"("id");
```

```
ALTER TABLE
```

```
"Action" ADD CONSTRAINT "action_item_type_id_foreign" FOREIGN  
KEY("item_type_id") REFERENCES "Item_type"("id");
```

SQL-запросы.

-- 1. Добавление новых пользователей в систему:

begin;

```
insert into Seller(id, name, region, category_1_id, system_name, inn, blocked)
  values(11948398, 'ООО "МОЯ ОБОРОНА"', '077', 62731,
'c444abdbfc0d7cf6b65b836c98ae991919e875397d951f69da9bf7b4e89b0d71',
'7708328899', false)
```

commit;

begin;

```
insert into Seller(id, name, region, category_1_id, system_name, inn, blocked)
  values(11948399, 'ООО "СОЮЗ СВЯТОГО ИОАННА ВОИНА"', '077',
52623,
'8626f45fc7e4060adab55e3a1cec939d4d2116cfaf084f18a1531910e9bf584b',
'7729705354', false)
```

commit;

begin;

```
insert into Seller(id, name, region, category_1_id, system_name, inn, blocked)
  values(11948400, 'ООО "3, 14-3-ДАТА"', '024', 9713,
'cebd886c166051bc8997e4df5b7316e9f8afa79a13055b668b406e9b584b2055',
'2455036974', false)
```

commit;

-- 2. Количество продаж у продавцов из Москвы, поставщики которых также находятся в Москве за ноябрь:

select

count(distinct action_id)

from Action a

inner join Seller se

on a.seller_id = se.id and se.region = '077'

inner join Supplier su

on a.supplier_id = su.id and su.region = '077'

where a.Date between '20221101' and '20221201'

-- 3. Проверка на совпадение категорий товаров и продавцов этих товаров:

```
select
  case
    when count(*) = 0 then true
    else false
  end as ok_status
from Action a
inner join Seller se
  on a.seller_id = se.id
inner join Item_type it
  on a.item_type_id = it.id
where it.category_1_id != se.category_1_id
```