

Санкт-Петербургский Национальный Исследовательский Университет
Информационных Технологий, Механики и Оптики

ПИиКТ

Курсовая работа
по дисциплине
«Информационные системы и базы данных»

Выполнили: Студенты группы Р33113

Мансуров Б.Б.

Кадыров А.Х.

Преподаватель: Николаев В.В.

Санкт-Петербург

2020г

Задание

Производство и продажа экзотических колбасных изделий

Что из себя представляет наша предметная область

Представим, что нам нужно проследить путь колбасных изделий начиная от цеха по производству до прилавков магазинов и разных мясных лавок.

Что из себя представляет наша предметная область: Есть много цехов где производят разные колбасные изделия у каждого цеха разный список продукции и каждый цех имеет свои фермы на разных городах и странах. В этих фермах есть разные животные включая экзотических животных. В зависимости от страны и города в фермах будут разводиться разные животные включая экзотических которые доступны только в этой местности. У каждого цеха имеется склад в которых хранятся готовые изделия в течение 5 суток. У каждого цеха есть специальные люди (сотрудники цеха) которые будут поставлять разные колбасные изделия клиентам, под клиентами подразумевается продавец в магазине или

в мясной лавке или еще где ни будь. Есть специальные лавки где продаются только экзотическая продукция цеха. У каждого цеха будет свой график, в зависимости от графика будут поставляться колбасные изделия. Сотрудник одновременно может быть сотрудников нескольких цехов. В зависимости от груза (колбасных изделий) каждому сотруднику будет предоставлена специальная машина для перевозки груза. Город разделен на участки, каждый сотрудник может поставлять колбасные изделия только на своих участках. Продавцы могут иметь несколько поставщиков из разных цехов. Бывают такие случаи, когда колбасные изделия могут иметь дефекты и в этом случае продавец может обратно возратить эти товары поставщикам, и поставщики в свою очередь цеху. Клиент не всегда платежа способен, и поставщики чтобы не потерять клиентов могут отдать в долг товары. Клиенты могут сделать специальные заказы которые будут обрабатываться отдельно. Для сотрудников цехов и постоянных клиентов будут предоставлены скидки и льготные услуги на продукцию. Есть каталог подписок для клиентов в зависимости от подписки клиент может составить список продукции, которых может заказать. Для VIP клиентов у нас будет VIP подписки заказы которых будут обрабатываться отдельно.

Процессы

Проследим путь заказа: Клиент приобретает подписку -> заполняет список продуктов для заказа -> заказ добавляется в график заказов -> проверяется наличия продукции на складе -> в зависимости от местоположения заказчика будет выбран поставщик -> заказ будет принят поставщиком -> и в зависимости от веса будет предоставлена машина для перевозки груза -> товар будет доставлен.

Проследим путь возврата: У клиента есть дефект -> он заполняет список дефектных продуктов, и он проверяется с его заказом и если в заказе есть такой продукт, то дефект принимается -> и в зависимости кто поставил этот продукт будет выбран поставщик -> поставщик принимает этот дефект -> и дефект предоставляется цеху.

Проследим путь VIP заказа: клиент вне зависимости от местоположения может сделать заказ -> и этот заказ попадает в отдельную таблицу спец заказов -> и если в складе каких страны, то есть городов в этой стране есть этот продукт -> то заказ будет принят ближайшим поставщиком -> и товар будет доставлен.

Покупка товара исп. скидку: клиент заполняет список продуктов для заказа -> проверяется есть ли скидка и тому подобное -> и если есть скидка зачитывается -> и дальше обрабатывается как обычный заказ.

1. Имеются цехи которые производят колбасные изделия
2. У каждого цеха имеется список колбасных изделий
3. Есть разные страны и разные города.
4. Имеется разные фермы где будут разводятся разные животные включая экзотических
5. У каждого цеха есть свои сотрудники которые будут поставлять колбасные изделия клиентам.
6. Есть список клиентов
7. Клиент может заказать колбасные изделия
8. Из заказов клиентов строится график доставок
9. В зависимости от заказа будет предоставлена машина для перевозки груза
10. Города разделены на участки, участки разделены между поставщиков, и поставщик

может поставлять колбасные изделия только на своем участок.

11. Если товар имеет дефекты клиент имеет право вернуть товар

1. Возврат от клиента поставщику
2. Поставщик цеху

12. Клиент может взять товар в долг

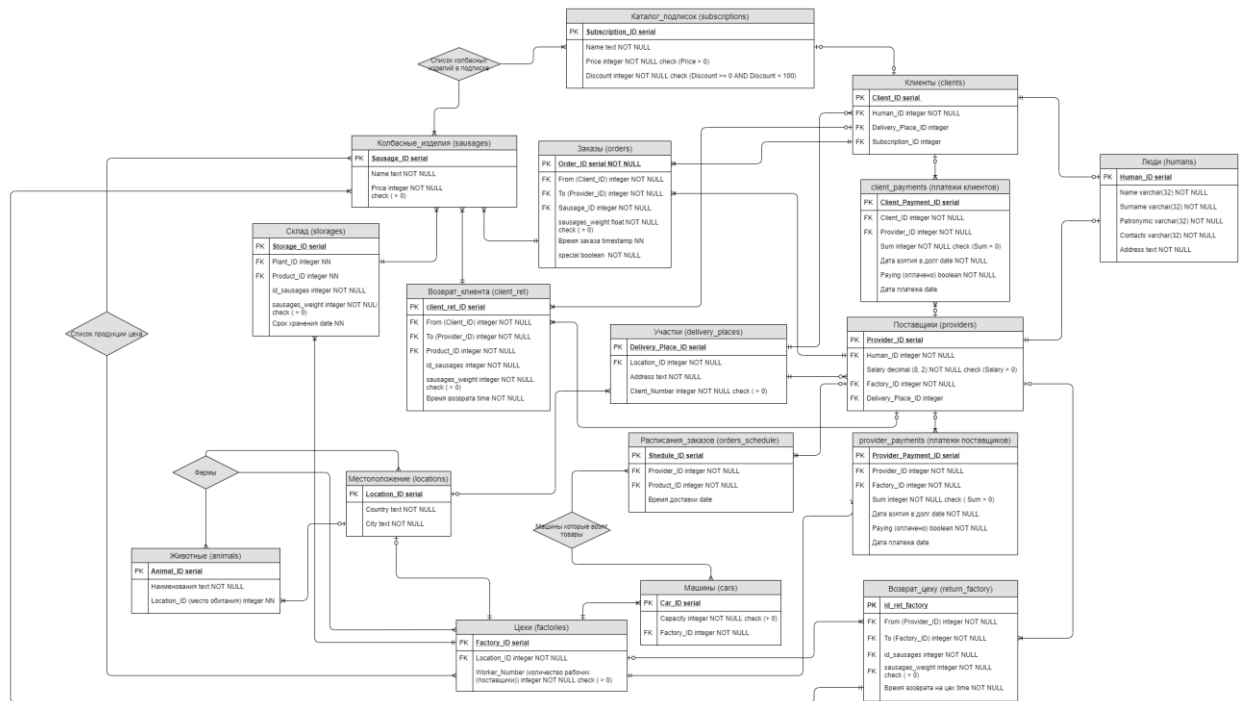
13. Есть специальные заказы которые будут обрабатываться отдельно

14. Есть каталог подписок и в зависимости от подписки будут составлено список продукции для заказа.

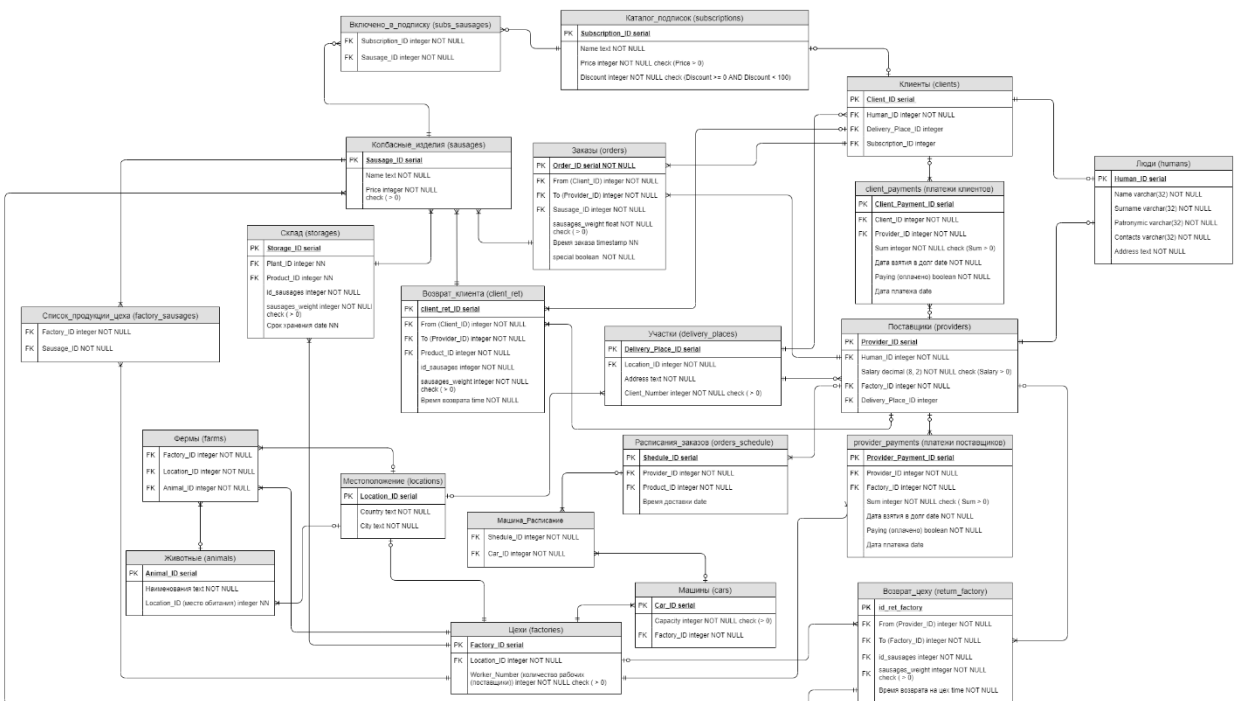
Список сущностей и их классификация

1. Цехи
2. Колбасные изделия
3. Животные
4. Местоположения
5. Склады
6. Клиенты (связь)
7. Поставщики
8. Заказ
9. Участки
10. Расписание
11. Машины
12. Подписки
13. Долги клиентов
14. Долги поставщиков
15. Возврат поставщикам
16. Возврат клиентам
17. Продукт

Инфологическая модель



Дата логическая модель



Реализация

```
CREATE TABLE humans
(
    id_human    serial,
    name        varchar(32) NOT NULL,
    surname     varchar(32) NOT NULL,
    patronymic  varchar(32) NOT NULL,
    contacts    varchar(32) NOT NULL,
    address     text        NOT NULL,
    PRIMARY KEY (id_human)
);

CREATE TABLE sausages
(
    id_sausage  serial,
    name        text        NOT NULL,
    price       decimal(7, 2) NOT NULL CHECK (price > 0),
    PRIMARY KEY (id_sausage)
);

CREATE TABLE locations
(
    id_location serial,
    country     text NOT NULL,
    city        text NOT NULL,
    PRIMARY KEY (id_location)
);

CREATE TABLE delivery_places
(
    id_delivery_place serial,
    id_location        integer NOT NULL,
    address            text    NOT NULL,
    client_num         integer NOT NULL CHECK ( client_num > 0),
    PRIMARY KEY (id_delivery_place),
    FOREIGN KEY (id_location) REFERENCES locations (id_location)
        ON UPDATE CASCADE
        ON DELETE CASCADE
);

CREATE TABLE factories
(
    id_factory    serial,
    location      integer NOT NULL,
    worker_num    integer NOT NULL CHECK (worker_num > 0),
    PRIMARY KEY (id_factory),
    FOREIGN KEY (location) REFERENCES locations (id_location)
);

CREATE TABLE animals
(
    id_animal    serial,
    name         text NOT NULL,
    location     integer NOT NULL,
    PRIMARY KEY (id_animal),
    FOREIGN KEY (location) REFERENCES locations (id_location)
);

CREATE TABLE cars
(
    id_car        serial,
    capacity       integer NOT NULL CHECK (capacity > 0),
    id_factory     integer NOT NULL,
```

```

        PRIMARY KEY (id_car),
        FOREIGN KEY (id_factory) REFERENCES factories (id_factory)
            ON UPDATE CASCADE
            ON DELETE CASCADE
    );

CREATE TABLE subscriptions
(
    id_subscription serial,
    name            text    NOT NULL,
    price           integer NOT NULL CHECK (price > 0),
    discount        integer NOT NULL CHECK (discount >= 0 AND discount <
100),
    PRIMARY KEY (id_subscription)
);

CREATE TABLE subs_sausages
(
    id_subscription integer NOT NULL,
    id_sausage      integer NOT NULL,
    PRIMARY KEY (id_subscription, id_sausage),
    FOREIGN KEY (id_subscription) REFERENCES subscriptions (id_subscription)
        ON UPDATE CASCADE
        ON DELETE CASCADE,
    FOREIGN KEY (id_sausage) REFERENCES sausages (id_sausage)
        ON UPDATE CASCADE
        ON DELETE CASCADE
);

CREATE TABLE providers
(
    id_provider      serial,
    id_human         integer NOT NULL,
    salary           decimal(8, 2) NOT NULL CHECK (salary > 0),
    id_factory       integer NOT NULL,
    delivery_place   integer,
    PRIMARY KEY (id_provider),
    FOREIGN KEY (id_human) REFERENCES humans (id_human)
        ON UPDATE CASCADE
        ON DELETE CASCADE,
    FOREIGN KEY (id_factory) REFERENCES factories (id_factory)
        ON UPDATE CASCADE
        ON DELETE CASCADE,
    FOREIGN KEY (delivery_place) REFERENCES delivery_places
(id_delivery_place)
        ON UPDATE CASCADE
        ON DELETE SET NULL
);

CREATE TABLE clients
(
    id_client        serial,
    id_human         integer NOT NULL,
    id_delivery_place integer,
    id_subscription  integer,
    PRIMARY KEY (id_client),
    FOREIGN KEY (id_human) REFERENCES humans (id_human)
        ON UPDATE CASCADE
        ON DELETE CASCADE,
    FOREIGN KEY (id_delivery_place) REFERENCES delivery_places
(id_delivery_place)
        ON UPDATE CASCADE
        ON DELETE SET NULL,

```

```

        FOREIGN KEY (id_subscription) REFERENCES subscriptions (id_subscription)
            ON UPDATE CASCADE
            ON DELETE SET NULL
    );

CREATE TABLE orders
(
    id_order serial,
    from      integer    NOT NULL,
    to        integer    NOT NULL,
    id_sausage integer NOT NULL,
    sausages_weight float NOT NULL CHECK (sausages_weight > 0),
    ord_time  timestamp NOT NULL,
    special   boolean NOT NULL,
    PRIMARY KEY (id_order),
    FOREIGN KEY (from) REFERENCES clients (id_client)
        ON UPDATE CASCADE
        ON DELETE CASCADE,
    FOREIGN KEY (to) REFERENCES providers (id_provider)
        ON UPDATE CASCADE
        ON DELETE CASCADE,
    FOREIGN KEY (id_sausage) REFERENCES sausages(id_sausage)

);

CREATE TABLE client_payments
(
    id_client_payment serial NOT NULL,
    id_client          integer NOT NULL,
    id_provider        integer NOT NULL,
    sum                integer NOT NULL CHECK (sum > 0),
    dept_date          date NOT NULL,
    paying             boolean NOT NULL,
    payment_date       date,
    PRIMARY KEY (id_client_payment),
    FOREIGN KEY (id_client) REFERENCES clients (id_client)
        ON UPDATE CASCADE
        ON DELETE CASCADE,
    FOREIGN KEY (id_provider) REFERENCES providers (id_provider)
        ON UPDATE CASCADE
        ON DELETE CASCADE
);

CREATE TABLE providers_payments
(
    id_provider_payment serial,
    id_provider          integer NOT NULL,
    id_factory           integer NOT NULL,
    sum                 integer NOT NULL CHECK (sum > 0),
    dept_date           date NOT NULL,
    paying              boolean NOT NULL,
    payment_date        date,
    FOREIGN KEY (id_provider) REFERENCES providers (id_provider)
        ON UPDATE CASCADE
        ON DELETE CASCADE,
    FOREIGN KEY (id_factory) REFERENCES factories (id_factory)
        ON UPDATE CASCADE
        ON DELETE CASCADE
);

CREATE TABLE order_shedule
(
    id_shedule serial,
    id_provider integer NOT NULL,

```

```

        id_sausage integer NOT NULL,
        sausages_weight float NOT NULL CHECK (sausages_weight > 0),
        del_time date NOT NULL,
        PRIMARY KEY (id_schedule),
        FOREIGN KEY (id_provider) REFERENCES providers (id_provider)
            ON UPDATE CASCADE
            ON DELETE CASCADE,
        FOREIGN KEY (id_sausage) REFERENCES sausages(id_sausage)
    );

CREATE TABLE car_schedule
(
    id_schedule integer NOT NULL,
    id_car integer NOT NULL,
    FOREIGN KEY (id_schedule) REFERENCES order_schedule (id_schedule)
        ON UPDATE CASCADE
        ON DELETE CASCADE,
    FOREIGN KEY (id_car) REFERENCES cars (id_car)
        ON UPDATE CASCADE
        ON DELETE CASCADE,
    PRIMARY KEY (id_schedule, id_car)
);

CREATE TABLE storages
(
    id_storage serial,
    id_factory integer NOT NULL,
    id_sausage integer NOT NULL,
    sausages_weight float NOT NULL CHECK (sausages_weight > 0),
    stor_life date,
    FOREIGN KEY (id_factory) REFERENCES factories (id_factory)
        ON UPDATE CASCADE
        ON DELETE CASCADE,
    FOREIGN KEY (id_sausage) REFERENCES sausages(id_sausage)
);

CREATE TABLE returns
(
    id_return serial,
    from integer NOT NULL,
    to integer NOT NULL,
    id_sausage integer NOT NULL,
    sausages_weight float NOT NULL CHECK (sausages_weight > 0),
    ret_time time NOT NULL,
    PRIMARY KEY (id_return),
    FOREIGN KEY (from) REFERENCES clients (id_client)
        ON UPDATE CASCADE
        ON DELETE CASCADE,
    FOREIGN KEY (to) REFERENCES providers (id_provider)
        ON UPDATE CASCADE
        ON DELETE CASCADE,
    FOREIGN KEY (id_sausage) REFERENCES sausages(id_sausage)
);

CREATE TABLE factory_sausages
(
    id_factory integer NOT NULL,
    id_sausage integer NOT NULL,
    PRIMARY KEY (id_factory, id_sausage),
    FOREIGN KEY (id_factory) REFERENCES factories (id_factory)
        ON UPDATE CASCADE
        ON DELETE CASCADE,

```



```

        FOREIGN KEY (id_sausage) REFERENCES sausages (id_sausage)
            ON UPDATE CASCADE
            ON DELETE CASCADE
    );

CREATE TABLE farms
(
    id_factory    integer NOT NULL,
    id_location   integer NOT NULL,
    id_animal     integer NOT NULL,
    PRIMARY KEY (id_factory, id_location, id_animal),
    FOREIGN KEY (id_factory) REFERENCES factories (id_factory)
        ON UPDATE CASCADE
        ON DELETE CASCADE,
    FOREIGN KEY (id_location) REFERENCES locations (id_location)
        ON UPDATE CASCADE
        ON DELETE CASCADE,
    FOREIGN KEY (id_animal) REFERENCES animals (id_animal)
        ON UPDATE CASCADE
        ON DELETE CASCADE
);

CREATE TABLE return_factory
(
    id_ret_factory serial,
    from            integer NOT NULL,
    to              integer NOT NULL,
    id_sausage      integer NOT NULL,
    sausages_weight float NOT NULL CHECK (sausages_weight > 0),
    ret_time        time     NOT NULL,
    PRIMARY KEY (id_ret_factory),
    FOREIGN KEY (from) REFERENCES providers (id_provider)
        ON UPDATE CASCADE
        ON DELETE CASCADE,
    FOREIGN KEY (to) REFERENCES factories (id_factory)
        ON UPDATE CASCADE
        ON DELETE CASCADE,
    FOREIGN KEY (id_sausage) REFERENCES sausages (id_sausage)
);

```