**Техническое задание на разработку информационной системы аптеки**

***1.Введение***

***1.1. Полное наименование системы***

Информационная система аптеки

***1.2. Общие сведения***

Система предназначена для повышения качества обработки данных и систематизации хранимой информации об аптеке и препаратах, целью создания которой является повышение эффективности работы аптеки.

***2.Основание для разработки***

Основанием для разработки программного продукта служит задание на курсовую работу “Аптека”.

***3. Назначение разработки***

Система предназначена для оперативного получения необходимой информации об аптеке, препаратах, работниках, заказах, клиентах и также может быть использована для ведения отчетности сотрудниками аптеки.

***4. Требования к программному изделию***

***4.1. Требования к функциональным характеристикам***

***4.1.1. Выполняемые функции***

***4.1.1.1. Для пользователя (фармацевт):***

* Введение новых клиентов
* Регистрация новых товаров
* Получение информации о наличии товара, о его количестве и сроках годности
* Изменение информации о клиентах
* Оформление заказов

***4.1.1.2. Для администратора:***

* Получение информации о сотрудниках
* Получение информации о выручке
* Введение в систему новых сотрудников
* Удаление сотрудников из системы
* Введение в систему новых поставщиков

***4.1.2. Требования к базе данных***

В базе данных должна храниться информация о:

* Клиентах
* Сотрудниках
* Товарах
* Поставщиках
* Заказах

***4.1.3. Требования к отчетам***

Система должна давать возможность получить следующие отчеты:

* Товарный чек
* Информация о выручке аптеки
* Информация о клиенте
* Информация о заказе (клиент и заказанный товар)
* Информация о товаре

***4.2. Требования к надежности***

Система должна обеспечить надежное хранение информации (защиту от сбоев, защиту от несанкционированного доступа).

***4.3. Требования к составу и параметрам технических средств***

Информационная система должна корректно работать на компьютере со следующими техническими характеристиками и выше:   
- Процессор Pentium V 2200 МГц  
- Оперативная память объемом 1024 Мб  
- Клавиатура  
- Манипулятор типа “мышь”  
- Устройство удаленного доступа к сети: модем или сетевая карта

***4.4. Специальные требования***

Специальные требования не предъявлялись.

***5. Требования к программной документации***

В состав документации входят пояснительная записка, руководство пользователя, альбом печатных форм.

***6. Порядок оформления и предъявления результатов работы***

Результаты представляются в виде отлаженной информационной системы, выполняющей функции данного ТЗ.

**Глава 1. Проектирование**

* 1. Анализ предметной области

Проанализируем техническое задание для понимания, какие бизнес-процессы будут реализованы в данной системе и какие данные должны находиться в базе данных.

* + 1. Товар

Таблица товар включает в себя все медикаменты, которые поступают на склад или заказываются клиентами, а также в данной таблице хранится информация и о других видах товаров, отпускаемых в аптеке.

* + 1. Заказы

Фармацевт создает заказ, полученный от клиента аптеки. В таблице «Заказ» содержится информация о заказанном товаре, его количестве, дате заказа, а также о клиенте, сделавшим заказ.

* + 1. Склад

Для учета товаров необходима таблица «Склад», в которой будет содержаться информация о наличии или отсутствии товара, его сроке годности и местоположении на складе.

* + 1. Доставка

Не всегда реализация заказа (отпуск товара клиенту) происходит при формировании нового заказа, поэтому была создана таблица «Доставка», которая будет хранить в себе информацию о заказе, о дате заказа, адресе получения и его получателе.

* + 1. Поставка

В данном разделе также необходимы таблицы «Поставщики» и «Поставка». Таблица «Поставка» хранит в себе информацию о закупаемом товаре, его поставщике и дате поставки.

Также у администратора есть возможность сформировать информацию о конкретной поставке в отчёт в бумажном виде.

* + 1. Сотрудники

В базе данных хранится информация о сотрудниках: ФИО, номер телефона, должность и зарплата.

* 1. **Бизнес процессы пользователей информационной системы**

Для выделения сущностей базы данных необходим анализ бизнес процессов пользователей.



Рис.1 БП «Формирование товарного чека»



Рис. 2 Диаграмма уникальных возможностей пользователя «Фармацевт»



Рис. 3 Диаграмма уникальных возможностей пользователя «Администратора»



Рис. 4 БП пользователя «Администратор»

* 1. **Выделение сущностей**

По бизнес процессам на рис. 1 - 4 можно выделить следующие сущности, с которыми работают фармацевт и администратор аптеки:

* Товарный чек
* Товар
* Доставка
* Склад
* Клиенты
* Поставка
* Заказ
* Сотрудники

Кроме того, для создания сервиса авторизации и аутентификации понадобятся отдельные сущности: «пользователь-роль» и «роль-страница» (для того, чтобы только определенный пользователь имел доступ только к определенным страницам).

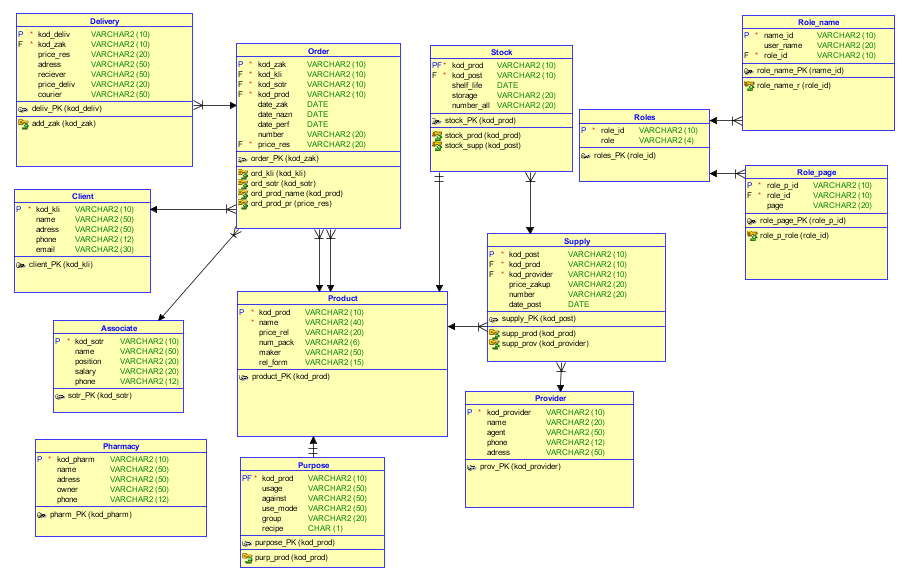
* 1. **Проектирование схем баз данных**

Затем, в таблицах «пользователь-роль» и «роль-страница» была обнаружена связь «много к много». В связи с этим была выделена отдельная таблица с ролями, чтобы получить две связи «один к много».

Во всех таблицах первичные ключи-суррогатные.

Кроме того, необходима отдельная таблица «Назначения», для того чтобы фармацевт мог определить к какой группе товаров относится то или иное лекарственное средство, его способ применения, показания к применению и противопоказанию.

На рисунке 5 представлена итоговая схема базы данных.

Рис. 5 Схема базы данных

**Приложения**

Скрипт, формирующий объекты БД

-- Generated by Oracle SQL Developer Data Modeler 18.1.0.082.1035

-- at: 2018-04-01 15:02:32 MSK

-- site: Oracle Database 11g

-- type: Oracle Database 11g

CREATE TABLE associate (

kod\_sotr VARCHAR2(10) NOT NULL,

name VARCHAR2(50),

position VARCHAR2(20),

salary VARCHAR2(20),

phone VARCHAR2(12)

);

ALTER TABLE associate ADD CONSTRAINT sotr\_pk PRIMARY KEY ( kod\_sotr );

CREATE TABLE client (

kod\_kli VARCHAR2(10) NOT NULL,

name VARCHAR2(50),

adress VARCHAR2(50),

phone VARCHAR2(12),

email VARCHAR2(30)

);

ALTER TABLE client ADD CONSTRAINT client\_pk PRIMARY KEY ( kod\_kli );

CREATE TABLE delivery (

kod\_deliv VARCHAR2(10) NOT NULL,

kod\_zak VARCHAR2(10) NOT NULL,

price\_res VARCHAR2(20),

adress VARCHAR2(50),

reciever VARCHAR2(50),

price\_deliv VARCHAR2(20),

courier VARCHAR2(50)

);

ALTER TABLE delivery ADD CONSTRAINT deliv\_pk PRIMARY KEY ( kod\_deliv );

CREATE TABLE "Order" (

kod\_zak VARCHAR2(10) NOT NULL,

kod\_kli VARCHAR2(10) NOT NULL,

kod\_sotr VARCHAR2(10) NOT NULL,

kod\_prod VARCHAR2(10) NOT NULL,

date\_zak DATE,

date\_nazn DATE,

date\_perf DATE,

"number" VARCHAR2(20),

price\_res VARCHAR2(20) NOT NULL

);

ALTER TABLE "Order" ADD CONSTRAINT order\_pk PRIMARY KEY ( kod\_zak );

CREATE TABLE pharmacy (

kod\_pharm VARCHAR2(10) NOT NULL,

name VARCHAR2(50),

adress VARCHAR2(50),

owner VARCHAR2(50),

phone VARCHAR2(12)

);

ALTER TABLE pharmacy ADD CONSTRAINT pharm\_pk PRIMARY KEY ( kod\_pharm );

CREATE TABLE product (

kod\_prod VARCHAR2(10) NOT NULL,

name VARCHAR2(40) NOT NULL,

price\_rel VARCHAR2(20),

num\_pack VARCHAR2(6),

maker VARCHAR2(50),

rel\_form VARCHAR2(15)

);

ALTER TABLE product ADD CONSTRAINT product\_pk PRIMARY KEY ( kod\_prod );

CREATE TABLE provider (

kod\_provider VARCHAR2(10) NOT NULL,

name VARCHAR2(20),

agent VARCHAR2(50),

phone VARCHAR2(12),

adress VARCHAR2(50)

);

ALTER TABLE provider ADD CONSTRAINT prov\_pk PRIMARY KEY ( kod\_provider );

CREATE TABLE purpose (

kod\_prod VARCHAR2(10) NOT NULL,

usage VARCHAR2(50),

against VARCHAR2(50),

use\_mode VARCHAR2(50),

"group" VARCHAR2(20),

recipe CHAR(1)

);

ALTER TABLE purpose ADD CONSTRAINT purpose\_pk PRIMARY KEY ( kod\_prod );

CREATE TABLE role\_name (

name\_id VARCHAR2(10) NOT NULL,

user\_name VARCHAR2(20),

role\_id VARCHAR2(10) NOT NULL

);

ALTER TABLE role\_name ADD CONSTRAINT role\_name\_pk PRIMARY KEY ( name\_id );

CREATE TABLE role\_page (

role\_p\_id VARCHAR2(10) NOT NULL,

role\_id VARCHAR2(10) NOT NULL,

page VARCHAR2(20)

);

ALTER TABLE role\_page ADD CONSTRAINT role\_page\_pk PRIMARY KEY ( role\_p\_id );

CREATE TABLE roles (

role\_id VARCHAR2(10) NOT NULL,

role VARCHAR2(4)

);

ALTER TABLE roles ADD CONSTRAINT roles\_pk PRIMARY KEY ( role\_id );

CREATE TABLE stock (

kod\_prod VARCHAR2(10) NOT NULL,

kod\_post VARCHAR2(10) NOT NULL,

shelf\_life DATE,

storage VARCHAR2(20),

number\_all VARCHAR2(20)

);

ALTER TABLE stock ADD CONSTRAINT stock\_pk PRIMARY KEY ( kod\_prod );

CREATE TABLE supply (

kod\_post VARCHAR2(10) NOT NULL,

kod\_prod VARCHAR2(10) NOT NULL,

kod\_provider VARCHAR2(10) NOT NULL,

price\_zakup VARCHAR2(20),

"number" VARCHAR2(20),

date\_post DATE

);

ALTER TABLE supply ADD CONSTRAINT supply\_pk PRIMARY KEY ( kod\_post );

ALTER TABLE delivery

ADD CONSTRAINT add\_zak FOREIGN KEY ( kod\_zak )

REFERENCES "Order" ( kod\_zak );

ALTER TABLE "Order"

ADD CONSTRAINT ord\_kli FOREIGN KEY ( kod\_kli )

REFERENCES client ( kod\_kli );

ALTER TABLE "Order"

ADD CONSTRAINT ord\_prod\_name FOREIGN KEY ( kod\_prod )

REFERENCES product ( kod\_prod );

ALTER TABLE "Order"

ADD CONSTRAINT ord\_prod\_pr FOREIGN KEY ( price\_res )

REFERENCES product ( kod\_prod );

ALTER TABLE "Order"

ADD CONSTRAINT ord\_sotr FOREIGN KEY ( kod\_sotr )

REFERENCES associate ( kod\_sotr );

ALTER TABLE purpose

ADD CONSTRAINT purp\_prod FOREIGN KEY ( kod\_prod )

REFERENCES product ( kod\_prod );

ALTER TABLE role\_name

ADD CONSTRAINT role\_name\_r FOREIGN KEY ( role\_id )

REFERENCES roles ( role\_id );

ALTER TABLE role\_page

ADD CONSTRAINT role\_p\_role FOREIGN KEY ( role\_id )

REFERENCES roles ( role\_id );

ALTER TABLE stock

ADD CONSTRAINT stock\_prod FOREIGN KEY ( kod\_prod )

REFERENCES product ( kod\_prod );

ALTER TABLE stock

ADD CONSTRAINT stock\_supp FOREIGN KEY ( kod\_post )

REFERENCES supply ( kod\_post );

ALTER TABLE supply

ADD CONSTRAINT supp\_prod FOREIGN KEY ( kod\_prod )

REFERENCES product ( kod\_prod );

ALTER TABLE supply

ADD CONSTRAINT supp\_prov FOREIGN KEY ( kod\_provider )

REFERENCES provider ( kod\_provider );

CREATE SEQUENCE sotr\_s START WITH 1 NOCACHE ORDER;

CREATE OR REPLACE TRIGGER sotr\_t BEFORE

INSERT ON associate

FOR EACH ROW

BEGIN

:new.kod\_sotr := sotr\_s.nextval;

END;

/

CREATE SEQUENCE kli\_s START WITH 1 NOCACHE ORDER;

CREATE OR REPLACE TRIGGER kli\_t BEFORE

INSERT ON client

FOR EACH ROW

BEGIN

:new.kod\_kli := kli\_s.nextval;

END;

/

CREATE SEQUENCE deliv\_s START WITH 1 NOCACHE ORDER;

CREATE OR REPLACE TRIGGER deliv\_t BEFORE

INSERT ON delivery

FOR EACH ROW

BEGIN

:new.kod\_deliv := deliv\_s.nextval;

END;

/

CREATE SEQUENCE zak\_s START WITH 1 NOCACHE ORDER;

CREATE OR REPLACE TRIGGER zak\_t BEFORE

INSERT ON "Order"

FOR EACH ROW

BEGIN

:new.kod\_zak := zak\_s.nextval;

END;

/

CREATE SEQUENCE pharm\_s START WITH 1 NOCACHE ORDER;

CREATE OR REPLACE TRIGGER pharm\_t BEFORE

INSERT ON pharmacy

FOR EACH ROW

BEGIN

:new.kod\_pharm := pharm\_s.nextval;

END;

/

CREATE SEQUENCE prod\_s START WITH 1 NOCACHE ORDER;

CREATE OR REPLACE TRIGGER prod\_t BEFORE

INSERT ON product

FOR EACH ROW

BEGIN

:new.kod\_prod := prod\_s.nextval;

END;

/

CREATE SEQUENCE prov\_s START WITH 1 NOCACHE ORDER;

CREATE OR REPLACE TRIGGER prov\_t BEFORE

INSERT ON provider

FOR EACH ROW

BEGIN

:new.kod\_provider := prov\_s.nextval;

END;

/

CREATE SEQUENCE purp\_prod\_s START WITH 1 NOCACHE ORDER;

CREATE OR REPLACE TRIGGER purp\_prod\_t BEFORE

INSERT ON purpose

FOR EACH ROW

BEGIN

:new.kod\_prod := purp\_prod\_s.nextval;

END;

/

CREATE SEQUENCE role\_n\_s START WITH 1 NOCACHE ORDER;

CREATE OR REPLACE TRIGGER role\_n\_t BEFORE

INSERT ON role\_name

FOR EACH ROW

BEGIN

:new.name\_id := role\_n\_s.nextval;

END;

/

CREATE SEQUENCE role\_p\_s START WITH 1 NOCACHE ORDER;

CREATE OR REPLACE TRIGGER role\_p\_t BEFORE

INSERT ON role\_page

FOR EACH ROW

BEGIN

:new.role\_p\_id := role\_p\_s.nextval;

END;

/

CREATE SEQUENCE role\_s START WITH 1 NOCACHE ORDER;

CREATE OR REPLACE TRIGGER role\_t BEFORE

INSERT ON roles

FOR EACH ROW

BEGIN

:new.role\_id := role\_s.nextval;

END;

/

CREATE SEQUENCE st\_prod\_s START WITH 1 NOCACHE ORDER;

CREATE OR REPLACE TRIGGER st\_prod\_t BEFORE

INSERT ON stock

FOR EACH ROW

BEGIN

:new.kod\_prod := st\_prod\_s.nextval;

END;

/

CREATE SEQUENCE post\_s START WITH 1 NOCACHE ORDER;

CREATE OR REPLACE TRIGGER post\_t BEFORE

INSERT ON supply

FOR EACH ROW

BEGIN

:new.kod\_post := post\_s.nextval;

END;

/