**Міністерство оствіти та науки України**

**Дніпропетровський національний університет залізничного транспорту**

**імені академіка В.Лазаряна**

**КУРСОВА РОБОТА**

Розробка реляційної бази даних, проектування структури сховища даних

з навчальної дисципліни «Бази даних»

Виконав:

студент групи 941

Валовый Б. Ю.

Науковий керівник:

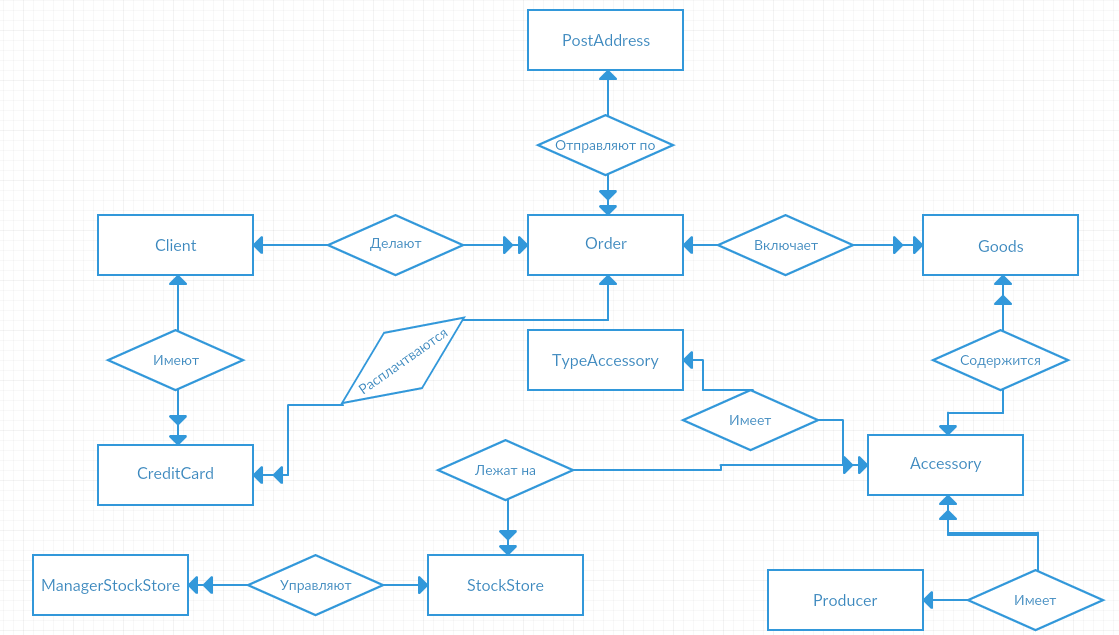
Пюра А. Н.

# Дніпро 2018

# 1 Постановка задачи, выбор предметной области, спецификация требований к управлению данными, определение основных типов запросов

Предметная область – модель интернет-магазина аксесуаров для ноутбуков и телефонов.

## Разработка информационно-логической и концептуальной EER модели БД



## 1.2 Определение типов сущностей, атрибутов, характеристик связей, установления возможных и первичных ключей

PK – primary key; FK – foreign key

**Client:**

* Id(primary key)
* First Name
* Last Name
* Phone number

**Order:**

* Id(PK)
* Id client(FK)
* Id post address(FK)

**Goods:**

* Id accessory(FK)
* Id order(FK)
* Accessory count

**Post address:**

* Id(PK)
* City
* Street
* Number of post department

**Producer:**

* Id(PK)
* Company name
* Country

**Accessory type:**

* Id(PK)
* Type name

**Accessory:**

* Id(PK)
* Name
* Color
* Type of link
* Id producer(FK)
* Id type(FK)

**StockStore:**

* StockId(PK)
* AccessoryTypeId(FK)
* City
* AddressOfStock

**StockStoreAccessory:**

* StockId(FK)
* AccessoryId(FK)
* PrimaryKey(StockId, AccessoryId)

**CreditCard:**

* CCNumb(PK)
* FirstName
* LastName
* ExpiryDate
* ValidationCode
* ClientId(FK)
* OrderId(FK)

**ManagerStockStore:**

* ManagerId(PK)
* FirstName
* LastName
* Phone
* StockId(FK)

## 1.3 Анализ требований к БД, выполнение процедур специализации / генерации, категоризации типов сущностей EER модели БД

## Правильно спроектированная БД должна удовлетворять следующим требованиям:

## 1. Минимальная избыточность. Непротиворечивость.

## 2. Целостность данных.

## 3. Независимость данных.

## 4. Возможность ведения (добавления и удаления) и актуализации (корректировки, модификации) данных.

## 5. Безопасность и секретность.

## 6. Высокая производительность. Минимальные затраты.

В разработанной базе данных были выделены следующей типы сущностей:

1. Client
2. Order
3. Goods
4. Post address
5. Producer
6. Accessory type
7. Accessory
8. CreditCard
9. StockStore
10. ManagerStockStore
11. StockStoreAccessory

# Разработка логического проекта базы данных

## 2.1 Разработка предварительной логической схемы БД на основе EER модели



## 2.2 Распределение не ключевых атрибутов, нормализация БД.

Описание процесса нормализации отношений базы данных: база данных была приведена к третей форме. В сущности Order были найдены транзитивные зависимости City -> Street -> Number of post и успешно нормализованы.



## 2.3 Обеспечение целостности БД

Целосность БД была обеспечена путем добавления к атрибутам таких ограничений: primary key, foreign key, check.

# Разработка физического проекта базы данных

## 3.1 Разработка структур таблиц и схемы БД с учетом возможностей СУБД. Скрипт создания БД

**TSQL:**

USE master;

GO

DROP DATABASE ShopDB;

GO

/\* Создание Базы Данных\*/

-------------------------------------------------------------------------

CREATE DATABASE ShopDB

ON

(

NAME = 'ShopDB',

FILENAME = 'G:\Studying\4\_course\DataBase\CourseWorkDB\DB\ShopDB.mdf',

SIZE = 200MB,

MAXSIZE = 1000MB,

FILEGROWTH = 100MB

)

LOG ON

(

NAME = 'LogShopDB',

FILENAME = 'G:\Studying\4\_course\DataBase\CourseWorkDB\DB\LogShopDB.ldf',

SIZE = 100MB,

MAXSIZE = 500MB,

FILEGROWTH = 50MB

)

COLLATE Cyrillic\_General\_CI\_AS -- Задаем кодировку для базы данных по умолчанию

-------------------------------------------------------------------------

-- Используем базу данных интернет-магазина

USE ShopDB

GO

-------------------------------------------------------------------------

/\* Создание таблиц \*/

-------------------------------------------------------------------------

CREATE SCHEMA Shop

GO

--таблица клиентов

CREATE TABLE Shop.Client

(

ClientId int IDENTITY(1,1) NOT NULL,

FirstName nvarchar(20) NOT NULL,

LastName nvarchar(20) NOT NULL,

Phone char(10) UNIQUE

)

-- Создание ограничения на телефон

ALTER TABLE Shop.Client

ADD CONSTRAINT CN\_ClientsPhone

-- Ограничение CHECK

CHECK (Phone LIKE '0[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]')

GO

-- Изменяем таблицу Client задав ограничение первичного ключа на столбце ClientId

ALTER TABLE Shop.Client

ADD CONSTRAINT PK\_Client

PRIMARY KEY (ClientId)

GO

--таблица заказов

CREATE TABLE Shop.Orders

(

OrderId int IDENTITY(1,1) NOT NULL

PRIMARY KEY,

ClientId int NOT NULL FOREIGN KEY REFERENCES Shop.Client(ClientId),

City nvarchar(30) NOT NULL,

Street nvarchar(30) NOT NULL,

NumberOfStreet nvarchar(10) NOT NULL

)

--таблица карточки для виплат клиента

CREATE TABLE Shop.CreditCard

(

CCNumb bigint NOT NULL PRIMARY KEY,

FirstName nvarchar(20) NOT NULL,

LastName nvarchar(20) NOT NULL,

ExpiryDate Date NOT NULL,

ClientId int NOT NULL FOREIGN KEY REFERENCES Shop.Client(ClientId),

OrderId int NULL FOREIGN KEY REFERENCES Shop.Orders(OrderId)

)

--производитель товаров

CREATE TABLE Shop.Producer

(

ProducerId int IDENTITY(1,1) NOT NULL

PRIMARY KEY,

CompanyName nvarchar(20),

Country nvarchar(20)

)

GO

--тип аксесуара

CREATE TABLE Shop.AccessoryType

(

AccessoryTypeId int IDENTITY(1,1) NOT NULL

PRIMARY KEY,

TypeName nvarchar(20) NOT NULL

)

--склады для аксесуара

CREATE TABLE Shop.StockStore

(

StockId int IDENTITY(1,1) NOT NULL PRIMARY KEY,

City nvarchar(30) NOT NULL,

AddressOfStock nvarchar(30) NOT NULL

)

--аксесуар

CREATE TABLE Shop.Accessory

(

AccessoryId int IDENTITY(1,1) NOT NULL

PRIMARY KEY,

AccessoryName nvarchar(20) NOT NULL,

Color nvarchar(20) NOT NULL,

TypeOfLink nvarchar(20) NOT NULL,

ProducerId int NOT NULL

FOREIGN KEY REFERENCES Shop.Producer(ProducerId),

AccessoryTypeId int NOT NULL

FOREIGN KEY REFERENCES Shop.AccessoryType(AccessoryTypeId)

)

GO

--многие ко многим StockStore - Accessory

CREATE TABLE Shop.StockStoreAccessory

(

StockId int NOT NULL

FOREIGN KEY REFERENCES Shop.StockStore(StockId),

AccessoryId int NOT NULL

FOREIGN KEY REFERENCES Shop.Accessory(AccessoryId),

PRIMARY KEY(StockId, AccessoryId)

)

GO

--менеджер склада

CREATE TABLE Shop.ManagerStockStore

(

ManagerId int IDENTITY(1,1) NOT NULL PRIMARY KEY,

FirstName nvarchar(20) NOT NULL,

LastName nvarchar(20) NOT NULL,

Phone char(10) UNIQUE,

StockId int NOT NULL

)

-- Создание ограничения на телефон

ALTER TABLE Shop.ManagerStockStore

ADD CONSTRAINT CN\_ManagerPhone

-- Ограничение CHECK

CHECK (Phone LIKE '0[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]')

GO

ALTER TABLE Shop.ManagerStockStore

ADD CONSTRAINT FK\_StockStoreId

FOREIGN KEY (StockId) REFERENCES Shop.StockStore(StockId)

GO

--товар

CREATE TABLE Shop.Goods

(

OrderId int NOT NULL,

AccessoryId int NOT NULL

FOREIGN KEY REFERENCES Shop.Accessory(AccessoryId),

CountAccessory smallint NOT NULL

)

GO

ALTER TABLE Shop.Goods

ADD CONSTRAINT FK\_OrderId

FOREIGN KEY (OrderId) REFERENCES Shop.Orders(OrderId)

GO