28

ДОДАТОК А

Графічний матеріал кваліфікаційної роботи

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

Метод застосування бізнес-аналітики для візуалізації фрагментів сховищ даних

Виконавець: ст. гр. СПм-22-3 РАДЧЕНКО І.В.

Керівник: ЗАВ. КАФЕДРИ КОВАЛЕНКО А.А.

АКТУАЛЬНІСТЬ РОЗРОБКИ

Дослідження в сфері візуалізації даних актуальне у зв'язку з швидким розвитком інформаційних технологій та постійним зростанням обсягів даних. Зрозуміння та ефективне управління цією інформацією є важливим для різних галузей, включаючи бізнес, науку та громадські сфери. Візуалізація даних стає ключовим інструментом для швидкого аналізу та прийняття обґрунтованих рішень. Отже, дослідження в цій області спрямоване на вдосконалення методів та інструментів візуалізації для відповіді на виклики інформаційного суспільства.

МЕТА РОБОТИ

Метою роботи є розробка моделі візуалізації даних та її дослідження, що має пришвидшити отримання та аналіз даних.

Для досягнення поставленої мети були поставлені такі задачі:

1. Розробка моделі мультибрендового магазину та бази даних для нього;
2. Заповнення бази даних магазину випадковими даними, наближеними до реального стану бізнесу;
3. Розробка Data Warehouse бази даних для мультибрендового магазину;
4. Розробка програм задля вивантаження, трансформацій та завантаження даних в систему Data Warehouse;
5. Розробка візуалізації даних за допомогою графічних звітів;

4

ОБ'ЄКТ ТА ПРЕДМЕТ ДОСЛІДЖЕННЯ

Об’єктом дослідження є процеси візуалізації даних. Предметом дослідження є засоби та методи структуризації, збереження та безпосередньо візуалізації інформації.

3

ОБРАНІ МЕТОДИ СТРУКТУРУВАННЯ ТА ЗБЕРЕЖЕННЯ ДАНИХ ТА ЇХ ВІЗУАЛІЗАЦІЇ

Data Warehouse - це централізована система зберігання і управління даними, призначена для аналізу та звітності. Основною метою Data Warehouse є інтеграція даних з різних джерел і їх трансформація в структуровану форму, що спрощує аналітичний доступ і використання

Business Intelligence (BI) – це ключова область для візуалізації даних. BI-платформи, такі як Tableau, Microsoft Power BI, та Qlik, надають зручний інтерфейс для створення дашбордів та звітів, які можуть взаємодіяти з різними джерелами даних. Вони відзначаються високою швидкістю розгортання та легкістю використання, спрощуючи процес аналізу для не-технічних користувачів.

5

ОБРАНИЙ ІНСТРУМЕНТАРІЙ

СУБД для розробки баз даних

Для генерації випадкових даних

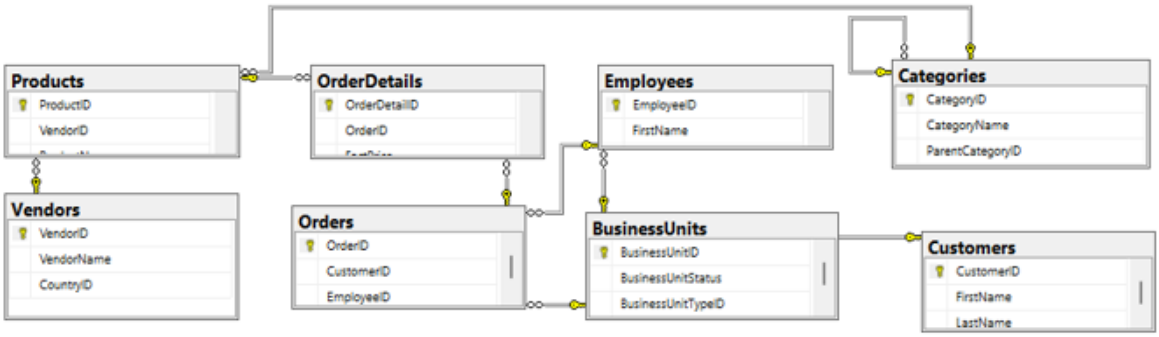
Застосунок для написання ETL програм

Середовище розробки баз данх

BI застосунок для візуалізації даних

6

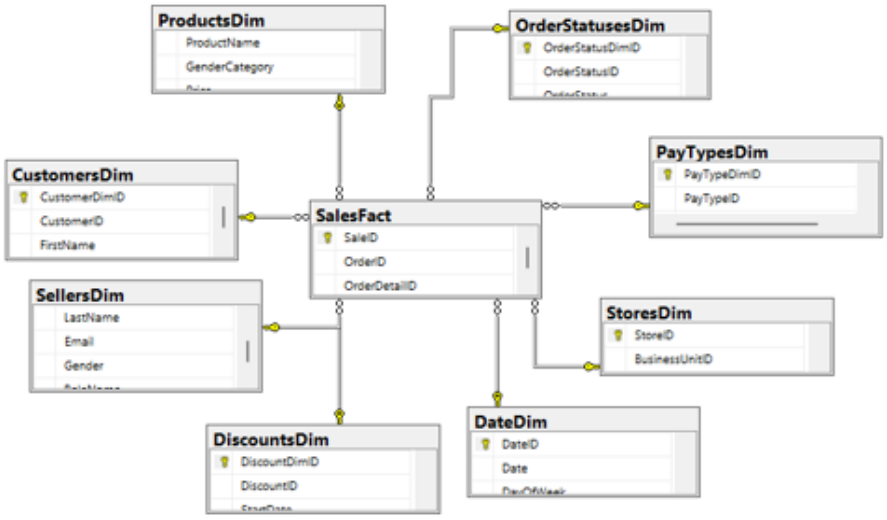
БАЗА ДАНИХ МУЛЬТИБРЕНДОВОГО МАГАЗИНУ



Головні сутності

7

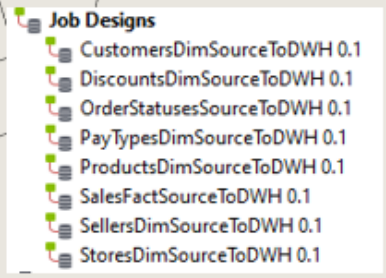
БАЗА ДАНИХ DATA WAREHOUSE



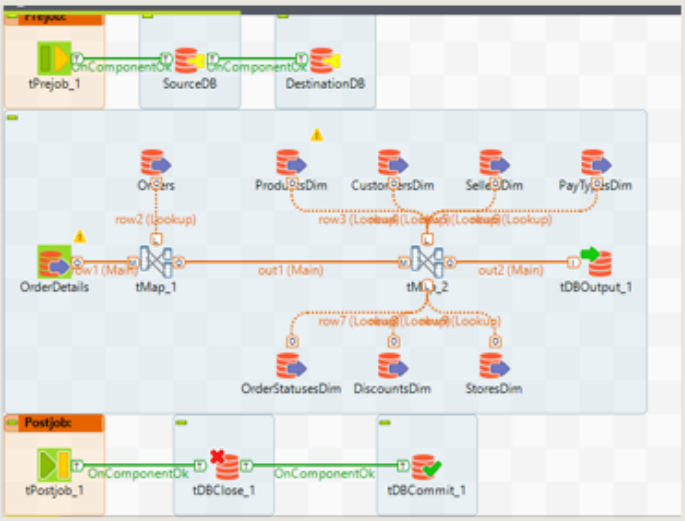
Повна схема бази даних

8

ETL ПРОГРАМИ ДЛЯ ТРАНСПОРТУВАННЯ ДАНИХ



Перелік усіх ETL програм



Графічні компоненти ETL програми "SalesFactSourceToDWH"

9

ДЕМОНСТРАЦІЯ: ВІЗУАЛІЗАЦІЯ ДАНИХ З POWER BI

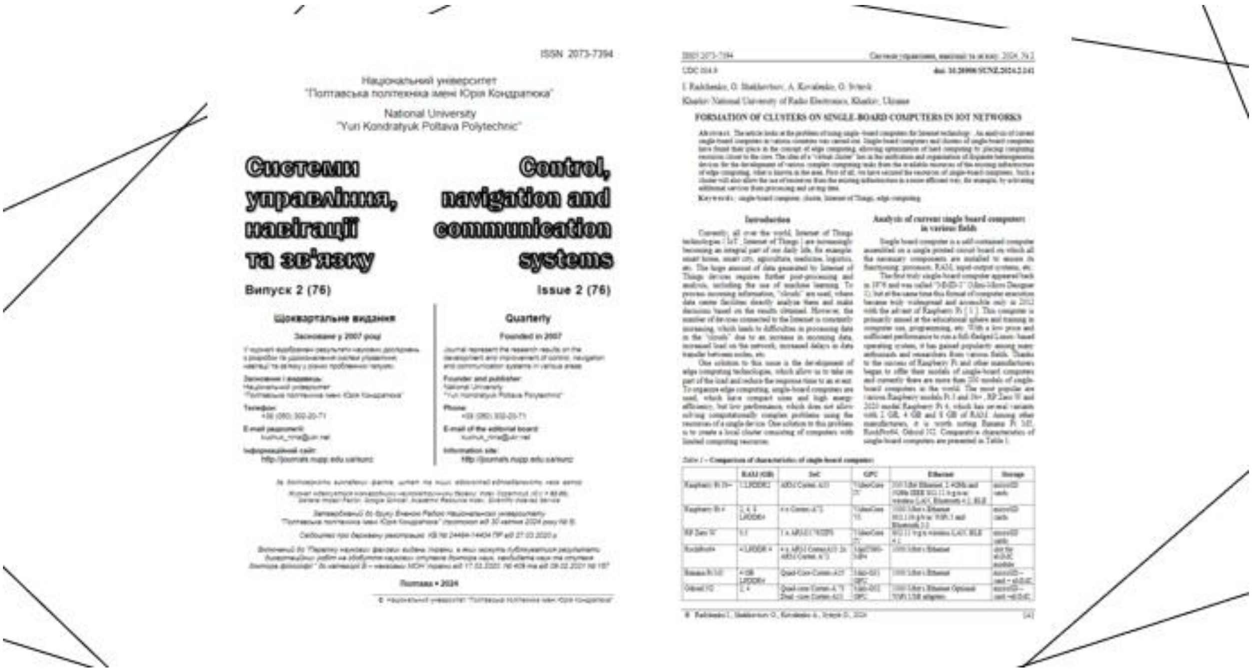
12

СТВОРЕНИЙ ГРАФІЧНИЙ ЗВІТ В POWER BI

Головна сторінка графічного звіту – “Revenue Summary”

ВИСНОВКИ

- Було розроблено модель мультибрендового магазину та базу даних для нього
- Було розроблено базу даних Data Warehouse, як централізоване сховище для зберігання аналітичних даних
- Було розроблено ETL програми для транспортування та трансформації даних
- Було розроблено графічний звіт з візуалізаціями даних



ISSN 2073-7394

Національний університет
"Полтавська політехніка імені Юрія Кондратюка"

National University
"Yuri Kondratyuk Poltava Polytechnic"

Системи управління, навігації та зв'язку

Control, navigation and communication systems

Випуск 2 (76)

Issue 2 (76)

Щоквартальне видання

Quarterly

Засноване у 2007 році

Founded in 2007

Полтава • 2024

FORMATION OF CLUSTERS ON SINGLE-BOARD COMPUTERS IN IOT NETWORKS

Introduction

Analysis of current single board computers in various fields

ДОДАТОК Б

Публікація

ISSN 2073-7394

Національний університет
"Полтавська політехніка імені Юрія Кондратюка"

National University
"Yuri Kondratyuk Poltava Polytechnic"

Системи управління, навігації та зв'язку

Control, navigation and communication systems

Випуск 2 (76)

Issue 2 (76)

Щоквартальне видання

Засноване у 2007 році

У журналі відображені результати наукових досліджень з розробки та удосконалення систем управління, навігації та зв'язку у різних проблемних галузях.

Засновник і видавець:
Національний університет
"Полтавська політехніка імені Юрія Кондратюка"

Телефон:
+38 (050) 302-20-71

E-mail редколегії:
kuchuk_nina@ukr.net

Інформаційний сайт:
http://journals.nupp.edu.ua/sunz

Quarterly

Founded in 2007

Journal represent the research results on the development and improvement of control, navigation and communication systems in various areas

Founder and publisher:
National University
"Yuri Kondratyuk Poltava Polytechnic"

Phone:
+38 (050) 302-20-71

E-mail of the editorial board:
kuchuk_nina@ukr.net

Information site:
http://journals.nupp.edu.ua/sunz

За достовірність викладених фактів, цитат та інших відомостей відповідальність несе автор

Журнал індексується міжнародними наукометричними базами: Index Copernicus (ICV = 82.05), General Impact Factor, Google Scholar, Academic Resource Index, Scientific Indexed Service

Затверджений до друку Вченою Радою Національного університету "Полтавська політехніка імені Юрія Кондратюка" (протокол від 30 квітня 2024 року № 5).

Свідоцтво про державну реєстрацію КВ № 24464-14404 ПР від 27.03.2020 р.

Включений до "Переліку наукових фахових видань України, в яких можуть публікуватися результати дисертаційних робіт на здобуття наукових ступенів доктора наук, кандидата наук та ступеня доктора філософії" до категорії Б – наказами МОН України від 17.03.2020 № 409 та від 09.02.2021 № 157

Полтава • 2024

ISSN 2073-7394 Системи управління, навігації та зв'язку. 2024. № 2

UDC 004.9 doi: 10.26906/SUNZ.2024.2.141

I. Radchenko, O. Shekhovtsov, A. Kovalenko, O. Sytnyk

Kharkiv National University of Radio Electronics, Kharkiv, Ukraine

FORMATION OF CLUSTERS ON SINGLE-BOARD COMPUTERS IN IOT NETWORKS

Abstract. The article looks at the problem of using single -board computers for Internet technology . An analysis of current single-board computers in various countries was carried out. Single-board computers and clusters of single-board computers have found their place in the concept of edge computing, allowing optimization of hard computing by placing computing resources closer to the core. The idea of a "virtual cluster" lies in the unification and organization of disparate heterogeneous devices for the development of various complex computing tasks from the available resources of the existing infrastructure of edge computing, what is known in the area. First of all, we have secured the resources of single-board computers. Such a cluster will also allow the use of resources from the existing infrastructure in a more efficient way, for example, by activating additional services from processing and saving data.

Keywords: single-board computer, cluster, Internet of Things, edge computing.

Introduction

Currently, all over the world, Internet of Things technologies (IoT , Internet of Things) are increasingly becoming an integral part of our daily life, for example: smart home, smart city, agriculture, medicine, logistics, etc. The huge amount of data generated by Internet of Things devices requires further post-processing and analysis, including the use of machine learning. To process incoming information, "clouds" are used, where data center facilities directly analyze them and make decisions based on the results obtained. However, the number of devices connected to the Internet is constantly increasing, which leads to difficulties in processing data in the "clouds" due to an increase in incoming data, increased load on the network, increased delays in data transfer between nodes, etc.

One solution to this issue is the development of edge computing technologies, which allow us to take on part of the load and reduce the response time to an event. To organize edge computing, single-board computers are used, which have compact sizes and high energy efficiency, but low performance, which does not allow solving computationally complex problems using the resources of a single device. One solution to this problem is to create a local cluster consisting of computers with limited computing resources.

Analysis of current single board computers in various fields

Single board computer is a self-contained computer assembled on a single printed circuit board on which all the necessary components are installed to ensure its functioning: processor, RAM, input-output systems, etc.

The first truly single-board computer appeared back in 1976 and was called "MMD-1" (Mini-Micro Designer 1), but at the same time this format of computer execution became truly widespread and accessible only in 2012 with the advent of Raspberry Pi [1]. This computer is primarily aimed at the educational sphere and training in computer use, programming, etc. With a low price and sufficient performance to run a full-fledged Linux- based operating system, it has gained popularity among many enthusiasts and researchers from various fields. Thanks to the success of Raspberry Pi and other manufacturers began to offer their models of single-board computers and currently there are more than 200 models of single-board computers in the world. The most popular are various Raspberry models Pi 3 and 3b+ , RP Zero W and 2020 model Raspberry Pi 4, which has several variants with 2 GB, 4 GB and 8 GB of RAM. Among other manufacturers, it is worth noting Banana Pi M5, RockPro64, Odroid N2. Comparative characteristics of single-board computers are presented in Table 1.

Table 1 – **Comparison of characteristics of single-board computers**

	RAM (GB)	SoC	GPU	Ethernet	Storage
Raspberry Pi 3b+	1 LPDDR2	ARM Cortex A53	VideoCore IV	330 Mbit Ethernet, 2.4GHz and 5GHz IEEE 802.11.b/g/n/ac wireless LAN, Bluetooth 4.2, BLE	microSD cards
Raspberry Pi 4	2, 4, 8 LPDDR4	4 x Cortex-A72	VideoCore VI	1000 Mbit/s Ethernet 802.11b/g/b/ac WiFi 5 and Bluetooth 5.0	microSD cards
RP Zero W	0.5	1 x ARM1176JZFS	VideoCore IV	802.11 b/g/n wireless LAN, BLE 4.1	microSD cards
RockPro64	4 LPDDR 4	4 x ARM CortexA53 2x ARM Cortex A72	MaliT860-MP4	1000 Mbit/s Ethernet	slot for eMMC module
Banana Pi M5	4 GB LPDDR4	Quad-Core Cortex-A55	Mali-G31 GPU	1000 Mbit/s Ethernet	microSD – card + eMMC
Odroid N2	2, 4	Quad-core Cortex-A 73 Dual -core Cortex-A53	Mali-G52 GPU	1000 Mbit/s Ethernet Optional WiFi USB adapters	microSD – card +eMMC

ДОДАТОК В

Текст програм та запитів

Запити на створення бази даних магазину:

```
CREATE TABLE Countries(
     CountryID int NOT NULL,  --id
     CountryName nvarchar(57) NOT NULL,  --name
     Region nvarchar(50) NULL, -- region
     Subregion nvarchar(50) NULL,
     Latitude numeric(19, 7) NULL,
     Longitude numeric(19, 7) NULL,
 CONSTRAINT PK_Countries_CountryID PRIMARY KEY (CountryID)
)

CREATE TABLE States(
     StateID int NOT NULL, --id
     StateName varchar(40) NOT NULL, --name
     CountryID int NOT NULL CONSTRAINT FK_States_Countries
FOREIGN KEY References Countries(CountryID), --countryid
     Latitude numeric(19, 7) NULL,
     Longitude numeric(19, 7) NULL,
 CONSTRAINT PK_States_StateID PRIMARY KEY (StateID)
 )

CREATE TABLE Cities(
     CityID int NOT NULL,
     CountryID int NOT NULL CONSTRAINT FK_Cities_Countries
FOREIGN KEY References Countries(CountryID),
     StateID int NOT NULL CONSTRAINT FK_Cities_States FOREIGN
KEY References States(StateID),
     CityName nvarchar(100) NOT NULL,
     Latitude numeric(19, 7) NULL,
     Longitude numeric(19, 7) NULL,
 CONSTRAINT PK_Cities_CityID PRIMARY KEY (CityID)
 )

CREATE TABLE Genders( --DONE обрати ієрархічну модель --
adjacency list
     GenderID int IDENTITY(1,1) NOT NULL,
     GenderName nvarchar(20) NOT NULL
 CONSTRAINT PK_Genders_GenderID PRIMARY KEY (GenderID)
 )

CREATE TABLE Categories( --DONE обрати ієрархічну модель --
adjacency list
     CategoryID int NOT NULL,
     CategoryName nvarchar(30) NOT NULL,
     ParentCategoryID int NULL CONSTRAINT
FK_Categories_Categories FOREIGN KEY References
Categories(CategoryID)
```

```
 CONSTRAINT PK_Categories_CategoryID PRIMARY KEY (CategoryID)
 )

CREATE TABLE Categories_Genders(
     CategoryID int NOT NULL CONSTRAINT
FK_Categories_Genders_CategoryID FOREIGN KEY References
Categories(CategoryID),
     GenderID int NOT NULL CONSTRAINT
FK_Categories_Genders_GenderID FOREIGN KEY References
Genders(GenderID)
)

 CREATE TABLE Vendors(
     VendorID int NOT NULL,
     VendorName nvarchar(50) NOT NULL,
     CountryID int NOT NULL,
 CONSTRAINT PK_Vendors_VendorID PRIMARY KEY (VendorID)
 )

CREATE TABLE PayTypes(
     PayTypeID int NOT NULL,
     PayTypeName nvarchar(20) NOT NULL,
 CONSTRAINT PK_PayTypes_PayTypeID PRIMARY KEY (PayTypeID)
 )

 CREATE TABLE Discounts(
     DiscountID int IDENTITY(1,1) NOT NULL,
     StartDate date NOT NULL, -- Old DateFrom
     EndDate date NOT NULL, -- Old DateTo
     DiscountPercent int NOT NULL,
     DiscountDescription nvarchar(200) NULL,
     DiscountStatus int NOT NULL,
 CONSTRAINT PK_Discounts_DiscountID PRIMARY KEY (DiscountID)
)

CREATE TABLE Colors(
     ColorID int IDENTITY(1,1) NOT NULL,
     ColorName nvarchar(50) NOT NULL,
 CONSTRAINT PK_Colors_ColorID PRIMARY KEY (ColorID)
)

CREATE TABLE Products(
     ProductID int NOT NULL,
     VendorID int NOT NULL CONSTRAINT FK_Products_Vendors
FOREIGN KEY References Vendors(VendorID),
     ProductName nvarchar(200) NOT NULL,
     GenderID int NOT NULL CONSTRAINT FK_Products_Genders
FOREIGN KEY References Genders(GenderID),
     Price money NOT NULL,
     NumImages smallint NOT NULL,
     Description nvarchar(4000) NOT NULL,
     ColorID int NULL CONSTRAINT FK_Products_Colors FOREIGN KEY
References Colors(ColorID),
```

```
     CategoryID int NOT NULL CONSTRAINT FK_Products_Categories
FOREIGN KEY References Categories(CategoryID),
 CONSTRAINT PK_Products_ProductID PRIMARY KEY (ProductID)
)

CREATE TABLE BusinessUnitTypes(
     BusinessUnitTypeID int NOT NULL,
     TypeName nvarchar(20) NOT NULL,
 CONSTRAINT PK_BusinessUnitType_BusinessUnitTypeID PRIMARY KEY
(BusinessUnitTypeID)
 )

CREATE TABLE BusinessUnits(
     BusinessUnitID int NOT NULL,
     BusinessUnitStatus bit NOT NULL,
     BusinessUnitTypeID int NOT NULL CONSTRAINT
FK_BusinessUnits_BusinessUnitTypes FOREIGN KEY References
BusinessUnitTypes(BusinessUnitTypeID),
 CONSTRAINT PK_BusinessUnits_BusinessUnitID PRIMARY KEY
(BusinessUnitID)
)

CREATE TABLE BusinessUnitAddresses(
     BusinessUnitID int NOT NULL,
     CountryID int NOT NULL CONSTRAINT FK_BUAddresses_Countries
FOREIGN KEY References Countries(CountryID),
     StateID int NOT NULL CONSTRAINT FK_BUAddresses_States
FOREIGN KEY References States(StateID),
     CityID int NOT NULL CONSTRAINT FK_BUAddresses_Cities
FOREIGN KEY References Cities(CityID),
     Street nvarchar(100) NOT NULL,
 CONSTRAINT PK_BusinessUnitAddress_BusinessUnitID PRIMARY KEY
(BusinessUnitID)
 )

CREATE TABLE Customers(
     CustomerID int NOT NULL,
     FirstName nvarchar(50) NOT NULL,
     LastName nvarchar(50) NOT NULL,
     BirthDate date NOT NULL,
     Gender nvarchar(10) NOT NULL,
     Email nvarchar(70) NOT NULL,
     PhoneNumber nvarchar(20) NOT NULL,
 CONSTRAINT PK_CustomerID PRIMARY KEY (CustomerID)
 )

CREATE TABLE Roles(
     RoleID int NOT NULL,
     RoleName nvarchar(30) NOT NULL,
     ManagerID int NULL CONSTRAINT FK_Roles_Roles FOREIGN KEY
References Roles(RoleID),
 CONSTRAINT PK_Roles_RoleID PRIMARY KEY (RoleID)
 )
```

```
CREATE TABLE Employees(
     EmployeeID int NOT NULL,
     FirstName nvarchar(50) NOT NULL,
     LastName nvarchar(50) NOT NULL,
     Gender nvarchar(10) NOT NULL,
     Email nvarchar(70) NOT NULL,
     PhoneNumber nvarchar(50) NOT NULL,
     RoleID int NOT NULL CONSTRAINT FK_Employees_Roles FOREIGN
KEY References Roles(RoleID),
     BusinessUnitID int NOT NULL CONSTRAINT
FK_Employees_BusinessUnits FOREIGN KEY References
BusinessUnits(BusinessUnitID),
 CONSTRAINT PK_Employees_EmployeeID PRIMARY KEY (EmployeeID)
 )

CREATE TABLE Invoices(
     InvoiceID int NOT NULL,
     InvoiceDate date NOT NULL,
     EmployeeID int NOT NULL CONSTRAINT FK_Invoices_Employees
FOREIGN KEY References Employees(EmployeeID),
 CONSTRAINT PK_Invoices_InvoiceID PRIMARY KEY (InvoiceID)
 )

 CREATE TABLE InvoiceDetails(
     InvoiceDetailID int NOT NULL,
     InvoiceID int NOT NULL CONSTRAINT
FK_InvoiceDetails_Invoices FOREIGN KEY References
Invoices(InvoiceID),
     ProductID int NOT NULL CONSTRAINT
FK_InvoiceDetails_Products FOREIGN KEY References
Products(ProductID),
     Quantity int NOT NULL,
 CONSTRAINT PK_InvoiceDetais_InvoiceDetailID PRIMARY KEY
(InvoiceDetailID)
 )

CREATE TABLE Shipments(
     ShipmentID int NOT NULL,
     BusinessUnitSourceID int NOT NULL,
     BusinessUnitDestinationID int NOT NULL,
     ShipmentDate datetime NOT NULL,
     ShipmentReceivedDate datetime NULL,
     InvoiceID int NOT NULL CONSTRAINT FK_Shipments_Invoices
FOREIGN KEY References Invoices(InvoiceID),
 CONSTRAINT PK_Shipments_InvoiceID PRIMARY KEY (ShipmentID)
 )

CREATE TABLE Supplies(
     SupplyID int NOT NULL,
     BusinessUnitID int NOT NULL CONSTRAINT
FK_Supplies_BusinessUnits FOREIGN KEY References
BusinessUnits(BusinessUnitID),
```

```
     VendorID int NOT NULL CONSTRAINT FK_Supplies_Vendors
FOREIGN KEY References Vendors(VendorID),
     SupplyDate datetime NOT NULL,
 CONSTRAINT PK_Supplies_SupplyID PRIMARY KEY (SupplyID)
)

CREATE TABLE SupplyDetails(
     SupplyDetailID int NOT NULL,
     SupplyID int NOT NULL CONSTRAINT FK_SupplyDetails_Supplies
FOREIGN KEY References Supplies(SupplyID),
     ProductID int NOT NULL CONSTRAINT FK_SupplyDetails_Products
FOREIGN KEY References Products(ProductID),
     Quantity int NOT NULL,
 CONSTRAINT PK_SupplyDetails_SupplyDetailID PRIMARY KEY
(SupplyDetailID)
)

CREATE TABLE OrderStatuses(
     OrderStatusID int NOT NULL,
     OrderStatus nvarchar(20) NOT NULL,
 CONSTRAINT PK_OrderStatuses_OrderStatusID PRIMARY KEY
(OrderStatusID)
)

CREATE TABLE Orders(
     OrderID int NOT NULL,
     CustomerID int NOT NULL CONSTRAINT FK_Orders_Customers
FOREIGN KEY References Customers(CustomerID),
     EmployeeID int NOT NULL CONSTRAINT FK_Orders_Employees
FOREIGN KEY References Employees(EmployeeID),
     PayTypeID int NOT NULL CONSTRAINT FK_Orders_PayTypes
FOREIGN KEY References PayTypes(PayTypeID),
     BusinessUnitID int NOT NULL CONSTRAINT
FK_Orders_BusinessUnits FOREIGN KEY References
BusinessUnits(BusinessUnitID),
     OrderDate datetime NOT NULL,
     Comment nvarchar(500) NULL,
     OrderStatusID int NOT NULL CONSTRAINT
FK_Orders_OrderStatuses FOREIGN KEY References
OrderStatuses(OrderStatusID),
 CONSTRAINT PK_Orders_OrderID PRIMARY KEY (OrderID)
 )

 CREATE TABLE OrderDetails(
     OrderDetailID int NOT NULL,
     OrderID int NOT NULL CONSTRAINT FK_OrderDetails_Orders
FOREIGN KEY References Orders(OrderID),
     FactPrice money NOT NULL,
     DiscountID int NOT NULL DEFAULT 0,
     ProductID int NOT NULL CONSTRAINT FK_OrderDetails_Products
FOREIGN KEY References Products(ProductID)
 CONSTRAINT PK_OrderDetails_OrderDetailID PRIMARY KEY
(OrderDetailID)
```

```
)
```

Частина запитів процедур генерації даних для бази даних магазину:

```
CREATE PROCEDURE OrdersDataGenerator
@SetStartDate DATETIME,
@SetEndDate DATETIME
AS
BEGIN
BEGIN TRY
     SET NOCOUNT ON;
          DECLARE
          @OrderID INT,
          @CustomerID INT,
          @EmployeeID INT,
          @PayTypeID INT,
          @UnitID INT,
          @OrderDate datetime,
          @Comment NVARCHAR(500),
          @OrderStatus INT,

          @StartDate DATETIME = @SetStartDate,
          @EndDate DATETIME = @SetEndDate,
          @Count INT,
          @Quantity INT,
          @Status INT

          SET @OrderID = isnull((
                                   select top (1) OrderID from
Orders order by OrderID desc
                              ), 0) + 1

          WHILE @StartDate<@EndDate
               BEGIN

               IF MONTH(@StartDate) BETWEEN 1 AND 3
               BEGIN
               SET @Quantity = rand() * 150 + 80;
               SET @Count = 0;
               END
               IF MONTH(@StartDate) BETWEEN 4 AND 6
               BEGIN
               SET @Quantity = rand() * 120 + 60;
               SET @Count = 0;
               END
               IF MONTH(@StartDate) BETWEEN 7 AND 9
               BEGIN
               SET @Quantity = rand() * 100 + 50;
               SET @Count = 0;
               END
               IF MONTH(@StartDate) BETWEEN 10 AND 12
               BEGIN
               SET @Quantity = rand() * 200 + 90;
               SET @Count = 0;
```

```
                END

                WHILE @Count<@Quantity
                      BEGIN

                      SELECT @CustomerID = (SELECT TOP(1)
CustomerID
                                                  FROM Customers
                                                  ORDER BY
NEWID());

                      SELECT @UnitID = (SELECT TOP(1)
BusinessUnitID
                                                FROM BusinessUnits
                                                WHERE
BusinessUnitStatus = 1 AND BusinessUnitTypeID = 2
                                                ORDER BY NEWID());

                      SELECT @PayTypeID = (SELECT TOP(1) PayTypeID
                                                  FROM PayTypes
                                                  WHERE
PayTypeID IN (1,2)
                                                  ORDER BY
NEWID());

                      SELECT @EmployeeID = (SELECT TOP(1)
EmployeeID
                                                  FROM Employees
                                                  WHERE RoleID =
2 and BusinessUnitID = @UnitID
                                                  ORDER BY
NEWID());

                      SELECT @OrderDate = @StartDate;

                      SELECT @Comment = 'Test comment'


                      SELECT @status = FLOOR(rand()*100+0)

                      IF (@status BETWEEN 0 AND 95)
                      BEGIN
                            SELECT @OrderStatus = 1
                      END
                      IF (@status BETWEEN 96 AND 100)
                      BEGIN
                            SELECT @OrderStatus = 2
                      END


                      --Insering data
                      INSERT INTO Orders(OrderID, CustomerID,
```

```
BusinessUnitID, EmployeeID, PayTypeID, OrderDate, Comment,
OrderStatusID)
                    VALUES (@OrderID, @CustomerID, @UnitID,
@EmployeeID, @PayTypeID, @OrderDate, @Comment, @OrderStatus);


                    SET @Count += 1
                    SET @OrderID += 1
                    END

               SET @StartDate += 1
               END


END TRY
BEGIN CATCH
     DECLARE @ErrorMessage NVARCHAR(4000);
     DECLARE @ErrorSeverity INT;
     DECLARE @ErrorState INT;

     SELECT
          @ErrorMessage = ERROR_MESSAGE(),
          @ErrorSeverity = ERROR_SEVERITY(),
          @ErrorState = ERROR_STATE();


     RAISERROR (@ErrorMessage, -- Message text.
                    @ErrorSeverity, -- Severity.
                    @ErrorState -- State.
               );
END CATCH;
END
```

Частина запитів на створення таблиць Data Warehouse бази даних:

```
CREATE TABLE DateDim(
     DateID int NOT NULL,
     [Date] date NOT NULL,
     [DayOfWeek] nvarchar(30) NOT NULL,
     [DayOfMonth] int NOT NULL,
     DayNumberOfWeek int NOT NULL,
     FullDateDescription nvarchar(30) NOT NULL,
     CalendarMonth nvarchar(10) NOT NULL,
     WeekOfYear int NOT NULL,
     [DayOfYear] int NOT NULL,
     HalfYear int NOT NULL,
     CalendarQuarter int NOT NULL,
     CalendarYear INT NOT NULL,
     HolidayIndicator nvarchar(150) NULL,
     CONSTRAINT PK_DateDim_DateID PRIMARY KEY (DateID)
)


CREATE TABLE StoresDim(
```

```
     StoreID int NOT NULL IDENTITY(1, 1),
     BusinessUnitID int NOT NULL,
     StoreName nvarchar(50) NOT NULL,
     StoreStatus nvarchar(10) NOT NULL,
     StoreTypeID int NOT NULL,
     StoreType nvarchar(20) NOT NULL,
     Country nvarchar(57) NOT NULL,
    [State] varchar(40) NOT NULL,
    City nvarchar(100) NOT NULL,
    Street nvarchar(100) NOT NULL,
     Latitude numeric(19, 7) NULL,
     Longitude numeric(19, 7) NULL,
     CONSTRAINT PK_StoresDim_StoreID PRIMARY KEY (StoreID)
)


CREATE TABLE SellersDim(
     SellerID int NOT NULL IDENTITY(1, 1),
     EmployeeID int NOT NULL,
     FirstName nvarchar(50) NOT NULL,
     LastName nvarchar(50) NOT NULL,
     Email nvarchar(70) NOT NULL,
     Gender nvarchar(10) NOT NULL,
     RoleName nvarchar(30) NOT NULL,
 CONSTRAINT PK_SellersDim_SellerID PRIMARY KEY (SellerID)
 )


 CREATE TABLE ProductsDim(
     ProductDimID int NOT NULL IDENTITY(1, 1),
     ProductID int NOT NULL,
     Brand nvarchar(50) NOT NULL,
     ProductName nvarchar(200) NOT NULL,
     GenderCategory nvarchar(20) NOT NULL,
     Price money NOT NULL,
     NumImages smallint NOT NULL,
     [Description] nvarchar(4000) NOT NULL,
     Color nvarchar(50) NOT NULL,
     Category nvarchar(30) NOT NULL,
     ParentCategory nvarchar(30) NOT NULL,
     IsActive nvarchar(10) NOT NULL,
 CONSTRAINT PK_ProductsDim_ProductDimID PRIMARY KEY
(ProductDimID)
)


 CREATE TABLE DiscountsDim(
     DiscountDimID int NOT NULL IDENTITY(1, 1),
     DiscountID int NOT NULL,
     StartDate date NOT NULL,
     EndDate date NOT NULL,
     DiscountPercent int NOT NULL,
     DiscountDescription nvarchar(200) NULL,
```

```
     DiscountStatus nvarchar(50) NULL,
     DaysDuration int NOT NULL,
 CONSTRAINT PK_DiscountsDim_DiscountID PRIMARY KEY
(DiscountDimID)
)


CREATE TABLE PayTypesDim(
     PayTypeDimID int NOT NULL IDENTITY(1, 1),
     PayTypeID int NOT NULL,
     PayTypeName nvarchar(20) NOT NULL,
 CONSTRAINT PK_PayTypes_PayTypeDimID PRIMARY KEY (PayTypeDimID)
 )

 CREATE TABLE CustomersDim(
     CustomerDimID int NOT NULL IDENTITY(1, 1),
     CustomerID int NOT NULL,
     FirstName nvarchar(50) NOT NULL,
     LastName nvarchar(50) NOT NULL,
     BirthDate date NOT NULL,
     Gender nvarchar(10) NOT NULL,
     Email nvarchar(70) NOT NULL,
 CONSTRAINT PK_CustomerID PRIMARY KEY (CustomerDimID)
 )

 CREATE TABLE OrderStatusesDim (
     OrderStatusDimID int NOT NULL IDENTITY(1, 1),
     OrderStatusID int NOT NULL,
     OrderStatus nvarchar(20) NOT NULL,
 CONSTRAINT PK_OrderStatuses_OrderStatusDimID PRIMARY KEY
(OrderStatusDimID)
 )


 CREATE TABLE SalesFact(
     SaleID int NOT NULL IDENTITY(1, 1),
     OrderID int NOT NULL,
     OrderDetailID int NOT NULL,
     ProductDimID int NOT NULL CONSTRAINT
FK_SalesFact_ProductsDim FOREIGN KEY References
ProductsDim(ProductDimID),
     CustomerDimID int NOT NULL CONSTRAINT
FK_SalesFact_CustomersDim FOREIGN KEY References
CustomersDim(CustomerDimID),
     SellerID int NOT NULL CONSTRAINT FK_SalesFact_EmployeesDim
FOREIGN KEY References SellersDim(SellerID),
     PayTypeDimID int NOT NULL CONSTRAINT
FK_SalesFact_PayTypesDim FOREIGN KEY References
PayTypesDim(PayTypeDimID),
     StoreID int NOT NULL CONSTRAINT FK_SalesFact_StoresDim
FOREIGN KEY References StoresDim(StoreID),
     OrderDateID int NOT NULL CONSTRAINT FK_SalesFact_DatesDim
FOREIGN KEY References DateDim(DateID),
```

```
     OrderStatusDimID int NOT NULL CONSTRAINT
FK_SalesFact_OrderStatusesDim FOREIGN KEY References
OrderStatusesDim(OrderStatusDimID),
     FactPrice money NOT NULL,
     RegularPrice money NOT NULL,
     DiscountDimID int NOT NULL CONSTRAINT
FK_SalesFact_DiscountsDim FOREIGN KEY References
DiscountsDim(DiscountDimID),
     CONSTRAINT PK_SalesFact_SaleID PRIMARY KEY (SaleID)
)
```