

# Практическое занятие 3

Создание хранилища данных, запросы к хранилищу данных

# Задание 1. Создание хранилища данных и последовательности (Сарка, стр. 46-52)

- Используя контекст базы данных master (USE master) создайте новую базу данных с именем TK463DW.
- Необходимые свойства базы данных:
  - один файл данных и один журнал транзакций на любом диске
  - начальный размер файла данных 300 Мб, автоувеличение 10 Мб.
  - начальный размер файла журнала 50 Мб, автоувеличение 10%.
- Создайте объект-последовательность с названием SeqCustomerDWKey. Начальное значение – 1, приращение – 1.
- Выполните создание измерений и таблицы фактов, описанных на последующих слайдах.
- Постройте диаграмму хранилища данных, сохраните ее как InternetSalesDW.

# SQL код создания базы данных (путь к файлу FILENAME может отличаться)

```
USE master;
IF DB_ID('TK463DW') IS NOT NULL
    DROP DATABASE TK463DW;
GO

CREATE DATABASE TK463DW
    ON PRIMARY
    (NAME = N'TK463DW', FILENAME = N'/var/opt/mssql/data/TK463DW.mdf',
    SIZE = 307200KB, FILEGROWTH = 10240KB)
    LOG ON
    (NAME = N'TK463DW_log', FILENAME = N'/var/opt/mssql/data/TK463DW_log.ldf',
    SIZE = 51200KB, FILEGROWTH = 10%);
GO

ALTER DATABASE TK463DW SET RECOVERY SIMPLE WITH NO_WAIT;
GO

USE TK463DW;
GO

CREATE SEQUENCE SeqCustomerDWKey AS INT
    START WITH 1
    INCREMENT BY 1;
GO
```

# Создание измерений\_1

Создайте измерение **Customers**, с источником информации из DimCustomer из хранилища данных AdventureWorksDW2019 с условиями, приведенными в таблице справа:

Имя столбца	Тип данных	Допустимость значений NULL	Замечания
CustomerDwKey	INT	NOT NULL	Суррогатный ключ; значения присваиваются с применением последовательности
CustomerKey	INT	NOT NULL	
FullName	NVARCHAR(150)	NULL	Сцепление имени и фамилии из DimCustomer
EmailAddress	NVARCHAR(50)	NULL	
BirthDate	DATE	NULL	
MaritalStatus	NCHAR(1)	NULL	
Gender	NCHAR(1)	NULL	
Education	NVARCHAR(40)	NULL	EnglishEducation из DimCustomer
Occupation	NVARCHAR(100)	NULL	EnglishOccupation из DimCustomer
City	NVARCHAR(30)	NULL	City из DimGeography
StateProvince	NVARCHAR(50)	NULL	StateProvinceName из DimGeography
CountryRegion	NVARCHAR(50)	NULL	EnglishCountryRegionName из DimGeography
Age	Inherited	Inherited	Вычисляемый столбец. Вычислите разницу между BirthDate и текущей датой и дискретизируйте результат, отнеся его к одной из трех групп: <ul style="list-style-type: none"><li>• если разница меньше либо равна 40, пометьте "Younger" (молодой);</li><li>• если разница больше 50, пометьте "Older" (пожилкой);</li><li>• в противном случае пометьте "Middle Age" (среднего возраста)</li></ul>
CurrentFlag	BIT	NOT NULL	По умолчанию 1

# SQL код создания измерения Customers

```
CREATE TABLE Customers
(
    CustomerDWKey INT NOT NULL,
    CustomerKey INT NOT NULL,
    FullName NVARCHAR (150) NULL,
    EmailAddress NVARCHAR (50) NULL,
    BirthDate DATE NULL,
    MaritalStatus NCHAR (1) NULL,
    Gender NCHAR (1) NULL,
    Education NVARCHAR (40) NULL,
    Occupation NVARCHAR (100) NULL,
    City NVARCHAR (30) NULL,
    StateProvince NVARCHAR (50) NULL,
    CountryRegion NVARCHAR (50) NULL,
    Age AS
        CASE
            WHEN BirthDate IS NULL THEN NULL
            WHEN DATEDIFF(yy,BirthDate,CURRENT_TIMESTAMP) > 50
            THEN 'Older'
            WHEN DATEDIFF(yy,BirthDate,CURRENT_TIMESTAMP) > 40
            THEN 'MiddleAge'
            ELSE 'Younger'
        END,
    CurrentFlag BIT NOT NULL DEFAULT 1,
    CONSTRAINT PK_Customers PRIMARY KEY (CustomerDWKey)
);
GO
```

# Создание измерений\_2

Создайте измерение **Products**, с источником информации из DimProducts и измерение **Dates** с источником информации DimDate из хранилища данных AdventureWorksDW2019 с условиями, приведенными в таблицах справа:

Имя столбца	Тип данных	Допустимость значений NULL	Замечания
ProductKey	INT	NOT NULL	
ProductName	NVARCHAR (50)	NULL	EnglishProductName из DimProduct
Color	NVARCHAR (15)	NULL	
Size	NVARCHAR (50)	NULL	
SubcategoryName	NVARCHAR (50)	NULL	EnglishProductSubcategoryName из DimProductSubcategory
CategoryName	NVARCHAR (50)	NULL	EnglishProductCategoryName из DimProductCategory

Имя столбца	Тип данных	Допустимость значений NULL	Замечания
DateKey	INT	NOT NULL	
FullDate	DATE	NOT NULL	FullDateAlternateKey из DimDate
MonthNumberName	NVARCHAR (15)	NULL	Сцепите MonthNumberOfYear (с ведущими нулями для номеров месяцев меньших 10) и EnglishMonthName из DimDate
CalendarQuarter	TINYINT	NULL	
CalendarYear	SMALLINT	NULL	

# SQL код создания измерений Products и Dates

```
CREATE TABLE Products
(
    ProductKey INT NOT NULL,
    ProductName NVARCHAR (50) NULL,
    Color NVARCHAR (15) NULL,
    Size NVARCHAR (50) NULL,
    SubcategoryName NVARCHAR (50) NULL,
    CategoryName NVARCHAR (50) NULL,
    CONSTRAINT PK_Products PRIMARY KEY (ProductKey)
);
GO
```

```
CREATE TABLE Dates
(
    DateKey INT NOT NULL,
    FullDate DATE NOT NULL,
    MonthNumberName NVARCHAR (15) NULL,
    CalendarQuarter TINYINT NULL,
    CalendarYear SMALLINT NULL,
    CONSTRAINT PK_Dates PRIMARY KEY (DateKey)
);
GO
```

# Создание таблицы фактов

Создайте таблицу фактов **InternetSales**, с источником информации из FactInternetSales из хранилища данных AdventureWorksDW2019 с условиями, приведенными в таблице справа:

Имя столбца	Тип данных	Допустимость значений NULL	Замечания
InternetSalesKey	INT	NOT NULL	IDENTITY(1,1)
CustomerDwKey	INT	NOT NULL	Используя бизнес-ключ CustomerKey из измерения Customers, найдите подходящее значение суррогатного ключа CustomerDwKey из измерения Customers
ProductKey	INT	NOT NULL	
DateKey	INT	NOT NULL	OrderDateKey из FactInternetSales
OrderQuantity	SMALLINT	NOT NULL	По умолчанию 0
SalesAmount	MONEY	NOT NULL	По умолчанию 0
UnitPrice	MONEY	NOT NULL	По умолчанию 0
DiscountAmount	FLOAT	NOT NULL	По умолчанию 0

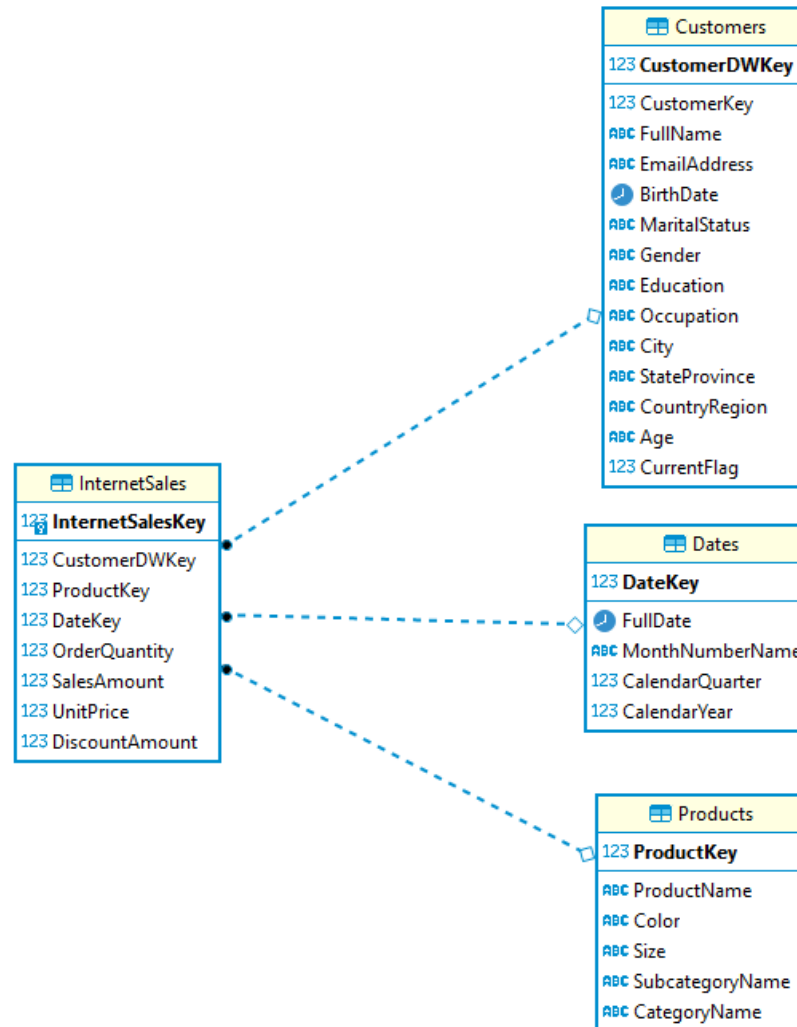


# SQL код создания таблицы фактов InternetSales

```
CREATE TABLE InternetSales
(
    InternetSalesKey INT NOT NULL IDENTITY (1,1),
    CustomerDWKey INT NOT NULL,
    ProductKey INT NOT NULL,
    DateKey INT NOT NULL,
    OrderQuantity SMALLINT NOT NULL DEFAULT 0,
    SalesAmount MONEY NOT NULL DEFAULT 0,
    UnitPrice MONEY NOT NULL DEFAULT 0,
    DiscountAmount FLOAT NOT NULL DEFAULT 0,
    CONSTRAINT PK_InternetSales PRIMARY KEY (InternetSalesKey)
);
GO
```

```
ALTER TABLE InternetSales ADD CONSTRAINT
    FK_InternetSales_Customers FOREIGN KEY (CustomerDWKey)
    REFERENCES Customers (CustomerDWKey);
ALTER TABLE InternetSales ADD CONSTRAINT
    FK_InternetSales_Products FOREIGN KEY (ProductKey)
    REFERENCES Products (ProductKey);
ALTER TABLE InternetSales ADD CONSTRAINT
    FK_InternetSales_Dates FOREIGN KEY (DateKey)
    REFERENCES Dates (DateKey);
GO
```

# Проверка схемы InternetSalesDW



## Задание 2. Загрузка хранилища данных (Сарка, стр. 60-69)

- Последовательно загрузите измерения Customers, Products, Dates данными из соответствующих таблиц из хранилища данных AdventureWorksDW2019 (DimCustomer, DimProduct, DimDate).
- Загрузите таблицу фактов InternetSales данными из соответствующих таблиц из хранилища данных AdventureWorksDW2019 (FactInternetSales).
- Выполните запрос к построенному хранилищу данных (приведен в слайдах ниже) и проанализируйте его результат.

# Код загрузки таблиц измерений

```
INSERT INTO Customers
    (CustomerDWKey, CustomerKey, FullName,
    EmailAddress, BirthDate, MaritalStatus,
    Gender, Education, Occupation,
    City, StateProvince, CountryRegion)
SELECT
    NEXT VALUE FOR SeqCustomerDWKey AS CustomerDWKey,
    C.CustomerKey,
    C.FirstName + ' ' + C.LastName AS FullName,
    C.EmailAddress, C.BirthDate, C.MaritalStatus,
    C.Gender, C.EnglishEducation, C.EnglishOccupation,
    G.City, G.StateProvinceName, G.EnglishCountryRegionName
FROM AdventureWorksDW2019.dbo.DimCustomer AS C
    INNER JOIN AdventureWorksDW2019.dbo.DimGeography AS G
    ON C.GeographyKey = G.GeographyKey;
GO
```

```
]INSERT INTO Products
    (ProductKey, ProductName, Color,
    Size, SubcategoryName, CategoryName)
SELECT
    P.ProductKey, P.EnglishProductName, P.Color,
    P.Size, S.EnglishProductSubcategoryName, C.EnglishProductCategoryName
FROM AdventureWorksDW2019.dbo.DimProduct AS P
    INNER JOIN AdventureWorksDW2019.dbo.DimProductSubcategory AS S
    ON P.ProductSubcategoryKey = S.ProductSubcategoryKey
    INNER JOIN AdventureWorksDW2019.dbo.DimProductCategory AS C
    ON S.ProductCategoryKey = C.ProductCategoryKey;
GO
```

```
]INSERT INTO Dates
    (DateKey, FullDate, MonthNumberName,
    CalendarQuarter, CalendarYear)
SELECT DateKey, FullDateAlternateKey,
    SUBSTRING (CONVERT (CHAR (8), FullDateAlternateKey, 112), 5, 2)
    + ' ' + EnglishMonthName,
    CalendarQuarter, CalendarYear
FROM AdventureWorksDW2019.dbo.DimDate;
GO
```

# Код загрузки таблицы фактов и запроса к готовому хранилищу

```
INSERT INTO InternetSales
    (CustomerDWKey, ProductKey, DateKey,
    OrderQuantity, SalesAmount,
    UnitPrice, DiscountAmount)
SELECT
    C.CustomerDWKey, FIS.ProductKey, FIS.OrderDateKey,
    FIS.OrderQuantity, FIS.SalesAmount,
    FIS.UnitPrice, FIS.DiscountAmount
FROM
    AdventureWorksDW2019.dbo.FactInternetSales AS FIS
    INNER JOIN Customers AS C
    ON FIS.CustomerKey = C.CustomerKey;
GO
```

```
SELECT C.CountryRegion, P.CategoryName, D.CalendarYear,
    SUM (I.SalesAmount) AS Sales
FROM InternetSales AS I
    INNER JOIN Customers AS C
    ON I.CustomerDWKey = C.CustomerDWKey
    INNER JOIN Products AS P
    ON I.ProductKey = P.ProductKey
    INNER JOIN Dates AS d
    ON I.DateKey = D.DateKey
GROUP BY C.CountryRegion, P.CategoryName, D.CalendarYear
ORDER BY C.CountryRegion, P.CategoryName, D.CalendarYear;
```