



**ТЕХНИЧЕСКИ УНИВЕРСИТЕТ –  
СОФИЯ**

---

**ФАКУЛТЕТ КОМПЮТЪРНИ СИСТЕМИ И  
УПРАВЛЕНИЕ**

**Курсов проект  
по  
SQL и МТД  
SQL Server Integration Services  
ВЪВ  
Visual Studio**

**Име:** Пламен Живков Прангов  
**Факултетен номер:** 121319028 (гр.222)

**София, 2022**

## **Съдържание:**

Задача .....	3
Изходната постановка .....	3
Използваните технологии .....	4
Реализацията .....	5
Източници .....	9

## Задача

В днешно време данните, които събираме са често разпръснати из различни системи/микросървиси или са в модел оптимизиран за достъп в реално време от клиенти или специфични задачи свързани с нормалната операция на предлаганата услуга. За един бизнес обаче историческата информация е от изключително значение и може да помогне за предвиждане на бъдещи действие, обучение на невронни мрежи, анализ поведението на потребителите и т.н. Лимит е само въображението за интерпретация на данните и често се налага агрегация на много източници, сложни трансформации и събирания спрямо различни критерии, което е невъзможно в текущата оперираща система, би отнело прекалено много време, което от своя страна би довело безполезни и остарели данни или би я натоварило прекалено много и би попречило на нормалните процеси. За целта използваме приложения, които поглъщат данните от различните източници и ги подготвят за анализ.

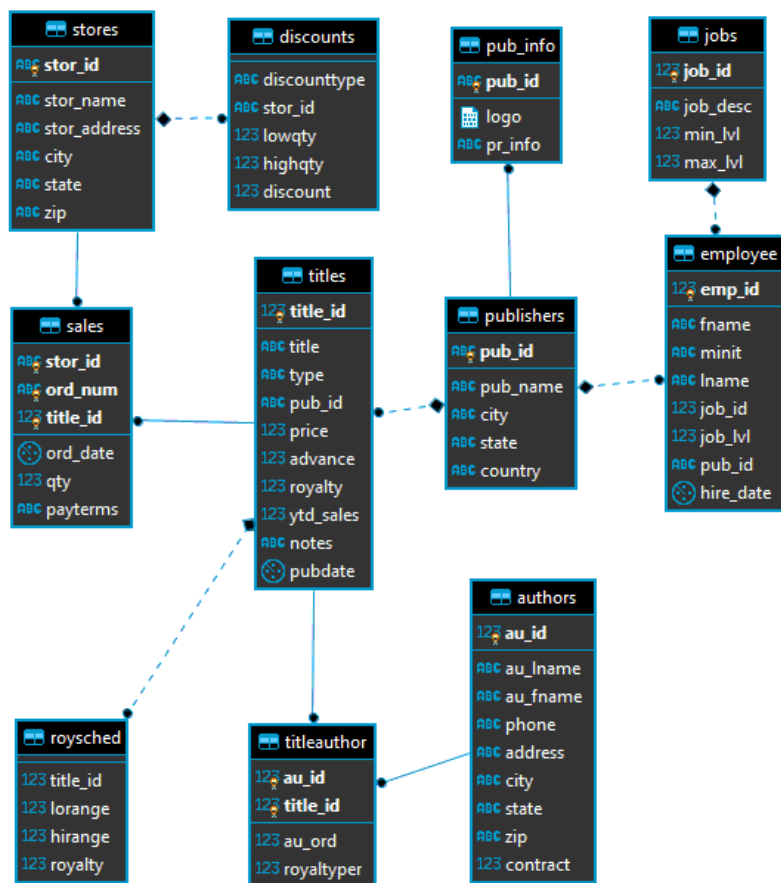
В нашата задача искаме да създадем проект, който да може да чете данни от един източник, да ги трансформира в различен модел и да ги въвежда директно в друг източник на данни или така наречената **ETL** операция (extract, transform, load). За целта ще използваме две схеми в **MS SQL Server** и **Integration Services Project** във **Visual Studio**, който използва **SQL Server Integration Services (SSIS)**[1] технологията вградена в **SQL Server** както и достъп до базите чрез стандартния **OleDb API**[2].

Пълните изисквания и стъпки могат да бъдат намерени в документ: УПР\_ETL-1.pdf

## Исходната постановка

Имаме релационна база съдържаща информация за верига книжарници:

- техните локации
- служители
- книги, които продават
- авторите на книгите
- издателските къщи, с които работят
- процентите на печалба при продажба
- намаления
- продажби



Фиг. 1 (ER диаграма на началната схема)

Поради, това че схемите са се променяли с времето имаме заглавия и продажби, за които данните са непълни.

За да анализираме данните и да извадим информация за това кои са най-добрите ни четиримесечия и най-продавани автори и издатели първо се нуждаем от това да нормализираме данните, да запълним пропуските и да ги запишем във формат лесен за четене.

## Използваните технологии

Microsoft SQL Server – СУБД в основата на проекта предоставяща достъп до ключови технологии улесняващи процеса на разработка. Някой от тях са:

- Windows Authentication – достъпа до базата се осъществява спрямо потребителите в операционната система (Windows), които от своя страна може да са създадени локално или да се контролират от ауторизационен сървър.
- OLEDB API – абстрактен интерфейс, който позволява кода ни да се свързва към бази от данни използвайки стандартни методи вместо директни SQL команди

- Integration Services (SSIS) Packages – Тези пакети могат да се качват и изпълняват директно на базата данни. Те могат да съдържат:
  - Контролни потоци (Control Flows) дефиниращи реда на събитията, които трябва да бъдат изпълнени
  - Потоци от данни (Data Flows) дефиниращи как се извличат, трансформират и зареждат данни.
  - Параметри – Статични променливи, които могат да се използват като форма на конфигурация спрямо различни среди и промяна на исканото поведение с цел да не тези неща да не бъдат записани директно в кода, където не могат да бъдат променяни толкова лесно или могат да бъдат пропуснати
  - Обработка на събития (Event Handlers) – Контролните потоци могат да дефинират определени действия при промяна състоянието на процеса. Това може да е обработка на грешки или допълнителни действия като изпращане на имейл при стартирани/приключване на задачи.

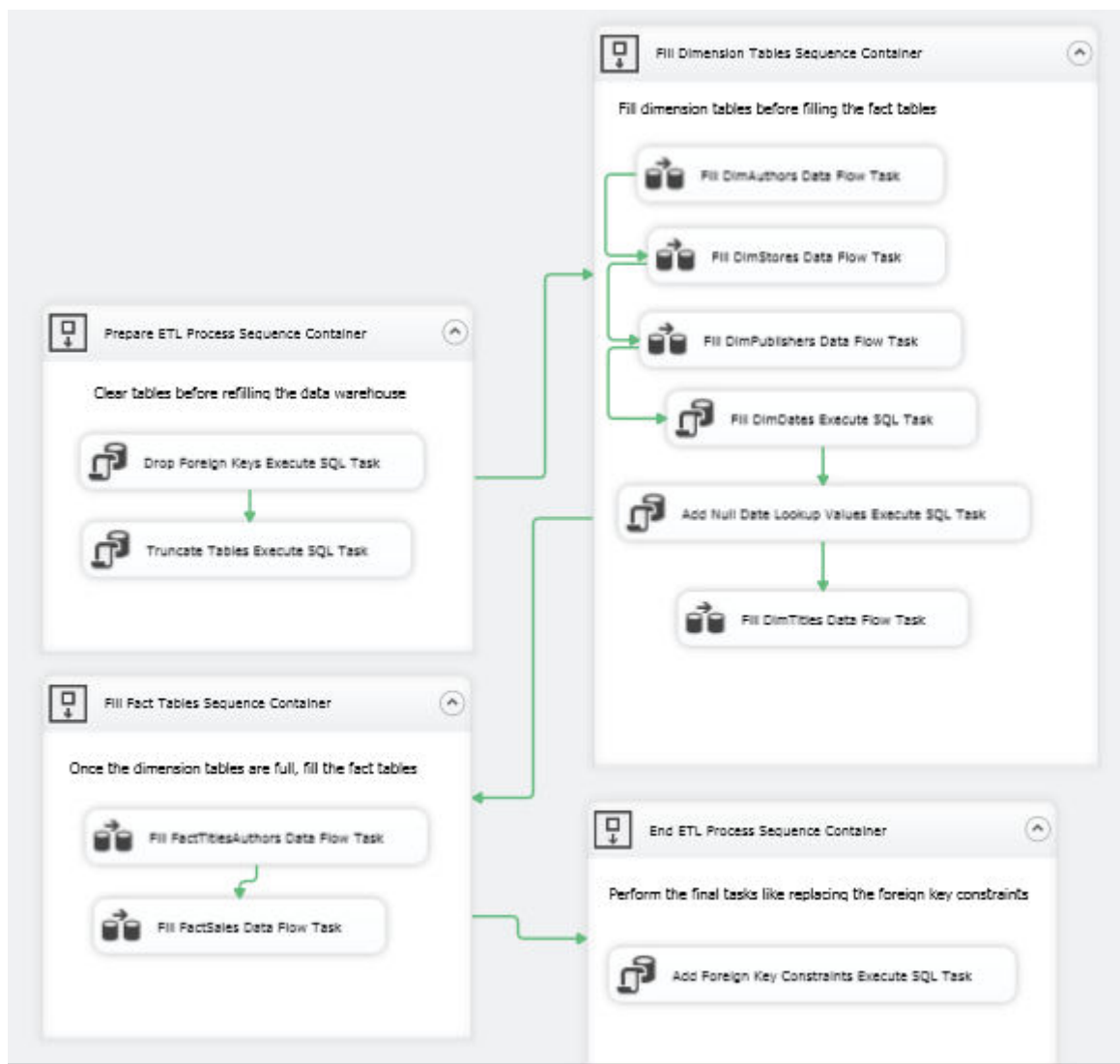
## Реализацията

Всички стъпки изискващи действие и визуална проверка могат да бъдат намерени под формата на снимки на екрана както следва:

- ЗАДАЧА 1: СЪЗДАВАНЕ НА SSIS ПРОЕКТ
  - /Задача1/ТА-[номер-на-задача] (2 изображения)
- ЗАДАЧА 2: КОНФИГУРИРАНЕ НА SSIS ПАКЕТА
  - ЗАДАВАНЕ НА СТРУКТУРАТА НА ПАКЕТА
    - /Задача2-А/ТВ-А-[номер-на-задача] (10 изображения)
  - КОНФИГУРИРАНЕ И ТЕСТВАНЕ НА CONTROL FLOW ЗАДАЧИТЕ
    - /Задача2-В/ТВ-В-[номер-на-задача] (16 изображения)
  - ЗАВЪРШВАНЕ НА ПАКЕТА
    - /Задача2-С/ТВ-С-[номер-на-задача] (41 изображения)
- ЗАДАЧА 3: ТЕСТВАНЕ НА SSIS ПАКЕТА
  - /Задача3/ТС-[номер-на-задача] (7 изображения)

Тук ще опишем само основните имплементационни детайли и резултати.

Като първа стъпка трябва да създадем **Integration Services Project** във **Visual Studio** с един Control Flow както е показано на фиг.2



В началото на контролния поток искаме да премахнем външните ключове, за да може да почистим таблиците без конфликти и да можем да инициализираме данните всеки път като стартираме процеса.

Последователно нормализираме и попълваме липсващите данните като извличаме данните от началната база “pubs” трансформираме ги посредством Data Flows и ги записваме в новосъздадената ни схема.

Например събиране името на автора в едно поле: `[AuthorName] = Cast( (au_fname + ' ' + au_lname) as nVarchar(100) )`

Ако имаме данни, които липсват като заглавия на книги ги запълваме с „Unknown” (Неизвестно), за да може да ги групираме по-лесно:

```
[TitleName] = Cast( isNull( [title], 'Unknown' ) as nvarchar(50) )
```

```
[TitleType] = Cast( isNull( [type], 'Unknown' ) as nvarchar(50) )
```

Цените и идентификатора за дата на публикация се приемат за „-1“ ако липсват като причина е вероятно, че подобна стойност е невъзможна в нормалните данни:

```
[TitlePrice] = Cast( isNull( [price], -1 ) as decimal(18, 4) )
```

```
[PublishedDateKey] = isNull([DWPubsSales].[dbo].[DimDates].[DateKey], -1)
```

Въз основа на новите ни таблици можем да извлечем лесно статистики и да ги запишем с нови таблици за бърз достъп вместо да ги изчисляваме наново

Например във таблица FactTitlesAuthors ще запишем колко поръчки е имала за всяка книга и нейния автор

```
SELECT [TitleKey] = DimTitles.TitleKey,  
       [AuthorKey] = DimAuthors.AuthorKey,  
       [AuthorOrder] = au_ord  
FROM pubs.dbo.titleauthor  
     JOIN DWPubsSales.dbo.DimTitles  
         On pubs.dbo.titleauthor.Title_id = DWPubsSales.dbo.DimTitles.TitleId  
     JOIN DWPubsSales.dbo.DimAuthors  
         On pubs.dbo.titleauthor.Au_id = DWPubsSales.dbo.DimAuthors.AuthorId
```

а в таблица FactSales нормализираме информацията за всяка поръчка. Кое е продаденото заглавие, датата на продажба, книжарница, автор и продадени бройки.

```
Select  [OrderNumber] = Cast(ord_num as nVarchar(50)),  
        [OrderDateKey] = DateKey,  
        [TitleKey] = DimTitles.TitleKey,  
        [StoreKey] = DimStores.StoreKey,  
        [SalesQuantity] = qty  
From pubs.dbo.sales  
     JOIN DWPubsSales.dbo.DimDates  
         On pubs.dbo.sales.ord_date = DWPubsSales.dbo.DimDates.date  
     JOIN DWPubsSales.dbo.DimTitles  
         On pubs.dbo.sales.Title_id = DWPubsSales.dbo.DimTitles.TitleId  
     JOIN DWPubsSales.dbo.DimStores  
         On pubs.dbo.sales.Stor_id = DWPubsSales.dbo.DimStores.StoreId
```

След изпълнение на контролния поток имаме всички подготвени данни за анализ.

На фигури 3А и 3Б имаме демонстрация за записаните резултати в новата ни база от данни

SQLQuery10.sql - ST...ent-PC(Student (52))\* X SQLQuery2.sql - STU...ent-PC

```

Select Top 100 * from dbo.DimAuthors
Select Top 100 * from dbo.DimStores
Select Top 100 * from dbo.DimPublishers
Select Top 100 * from dbo.DimDates
Select Top 100 * from dbo.DimTitles
Select Top 100 * from dbo.FactSales

```

.00 %

Results Messages

	AuthorKey	AuthorId	AuthorName	AuthorState
1	1	172-32-1176	Johnson White	CA
2	2	213-46-8915	Marjorie Green	CA
3	3	238-95-7766	Cheryl Carson	CA
4	4	267-41-2394	Michael O'Leary	CA
5	5	274-80-9391	Dean Straight	CA
6	6	341-22-1782	Meander Smith	KS
7	7	409-56-7008	Abraham Bennet	CA
8	8	427-17-2319	Ann Dull	CA

	StoreKey	StoreId	StoreName
1	1	6380	Eric the Read Books
2	2	7066	Barnum's
3	3	7067	News & Brews
4	4	7131	Doc-U-Mat: Quality...
5	5	7896	Fricative Bookshop
6	6	8042	Bookbeat

	PublisherKey	PublisherId	PublisherName
1	1	0736	New Moon Books
2	2	0877	Binnet & Hardley
3	3	1389	Algodata Infosy...
4	4	1622	Five Lakes Publ...
5	5	1756	Ramona Publish...
6	6	9901	GGG&G

Фиг. 3А резултати от контролния поток

	TitleKey	TitleId	TitleName	TitleType	PublisherKey	TitlePrice	PublishedDateKey
1	1	BU1111	Cooking with Computers: Surreptitious Balance Shee	business	3	11.9500	525
2	2	MC2222	Silicon Valley Gastronomic Treats	mod_cook	2	19.9900	525
3	3	BU1032	The Busy Executive's Database Guide	business	3	19.9900	528
4	4	PS3333	Prolonged Data Deprivation: Four Case Studies	psycholo...	1	19.9900	528
5	5	PS7777	Emotional Security: A New Algorithm	psycholo...	1	7.9900	528
6	6	TC4203	Fifty Years in Buckingham Palace Kitchens	trad_cook	2	11.9500	528
7	7	TC7777	Sushi, Anyone?	trad_cook	2	14.9900	528
8	8	PS2091	Is Anger the Enemy?	psycholo...	1	10.9500	531

	OrderNumber	OrderDateKey	TitleKey	StoreKey	SalesQuantity
1	423LL922	1718	9	6	15
2	423LL930	1718	3	6	10
3	6871	1718	3	1	5
4	722a	1717	8	1	3
5	A2976	1240	16	2	50
6	D4482	1718	8	3	10
7	N914008	1718	8	4	20
8	N914014	1718	9	4	25

	TitleKey	AuthorKey	AuthorOrder
1	1	4	2
2	1	17	1
3	2	14	1

Фиг. 3Б резултати от контролния поток



## Източници

Информацията е смислено събрана и обобщена от следните източници:

[1] [https://en.wikipedia.org/wiki/SQL\\_Server\\_Integration\\_Services](https://en.wikipedia.org/wiki/SQL_Server_Integration_Services)

[2] [https://en.wikipedia.org/wiki/OLE\\_DB](https://en.wikipedia.org/wiki/OLE_DB)

[3] <https://docs.microsoft.com/en-us/sql/integration-services/sql-server-integration-services?view=sql-server-ver16> (цялостна информация за SSIS и използването им в SQL Server)

[4] <https://www.ibm.com/cloud/learn/etl> (цялостна информация за ETLs и използването им)