SSRS ilə hesabatlılıq: Ümumi gündəlik hesabat

SSRS ilə tanışlıq üçün mən yalnız mənim təxəyyülümdə olan IT sahəsində xidmətlər ilə məşğul olan “ITSER” şirkətinin gündəlik satış hesabatını hazırlayacam.

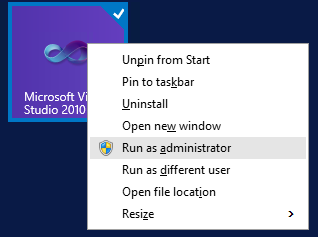
**Problem**: ITSER-in satış şöbəsinin işçiləri hər gün, gün ərzində baş vermiş satışlar haqqında məlumatı satış müdiri və direktora göndərirlər. Hesabatda rəqəmlər bəzən dəqiq olmur, hazırlayan şəxsdən aslı olaraq formatı və visual görüntüsü dəyişir, və ya ümumiyyətlə insan faktoru üzündən gəlmir.

**Hədəf**: Unikal, agregassiya olunmuş hesabat yaradıb cədvəl üzrə lazimi ünvanlara göndərmək.

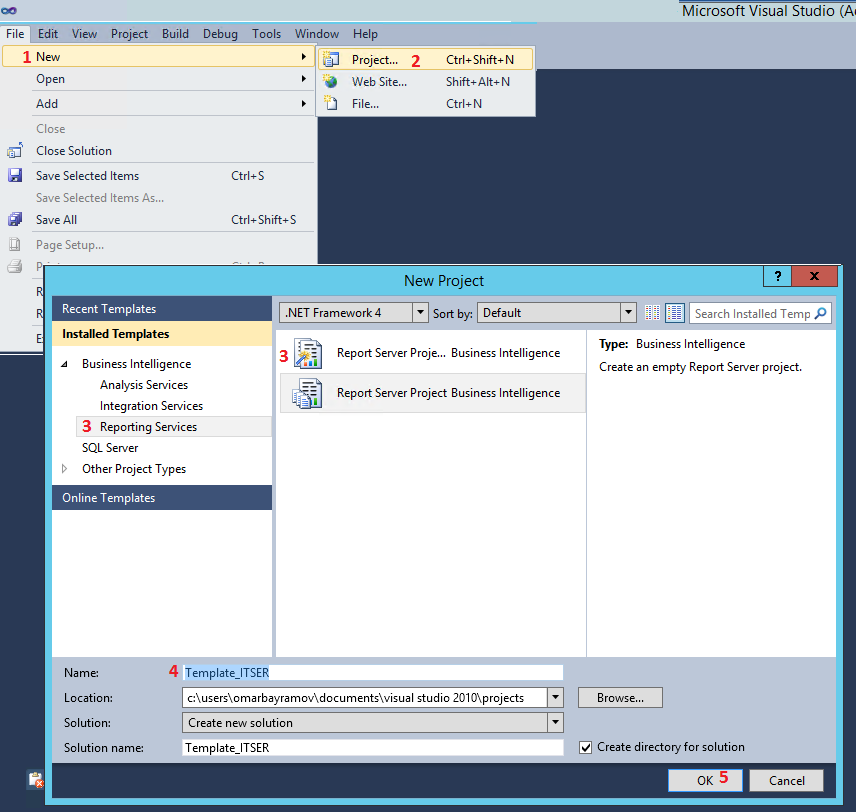
**İstifadə olunan program təminatı**:

1. Microsoft SQL Server 2012(RDMS, Reporting Services)
2. Microsoft Visual Studio 2010

Başlayaq!

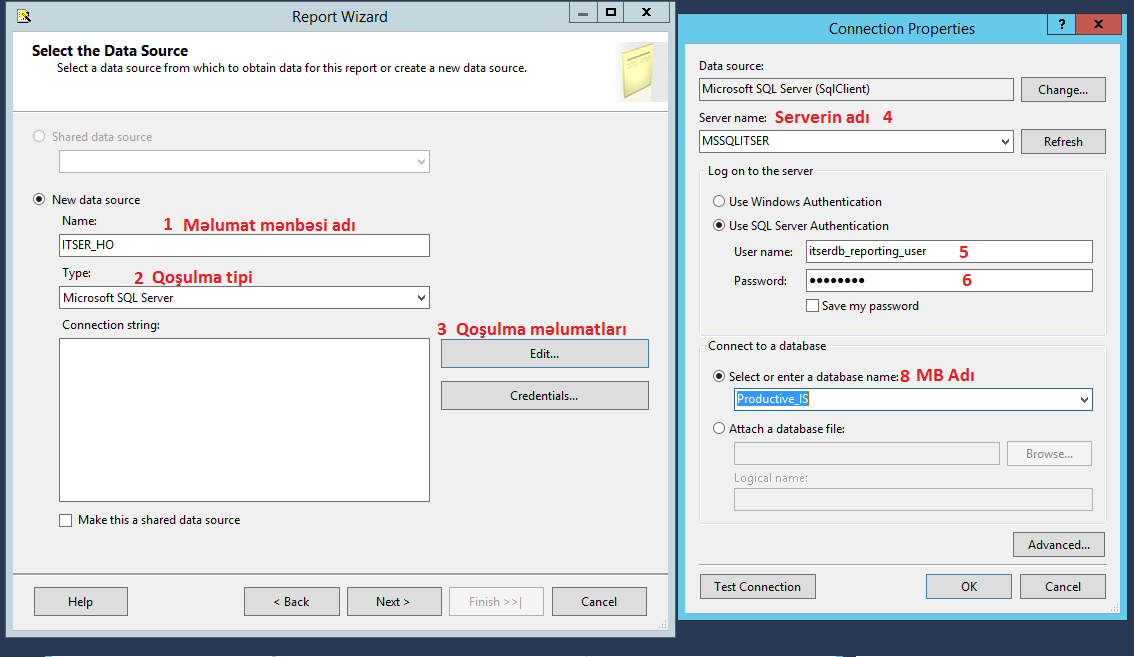
İlk öncə “**Start**” menyusunu daxil olaraq “**Microsoft Visual Studio 2010**”-u Administrator adınan açırıq.

Açılacaq pəncərədə “**File”🡪Project🡪Reporting Services🡪Reporting Server Project Wizard**” seçirik.

Düzü adətən mən daha çox “Reporting Server Project” seçib məzmunu bir-bir doldurmağı tərciyə edirəm, amma bu təlimi çətinləşdirməmək üçün “Köməkçi”-dən istifadə edəcəyik.

Növbəti pəncərədə öyrəşdiyimiz menyular ilə qarşılacağsınız.

İlk öncə biz Məlumat Mənbəsi yaratmalıyıq və ona ad verməliyik. Mən ona “**ITSER\_HO**” adı verdim. Qoşulacağımız MBİS *Microsoft SQL Server*-dir. Seçib *Edit* vasitəsi ilə qoşulma məlumatların yeni açılacaq pəncərəyə daxil edib *OK*  və *OK* basıb davam edirik.



Növbəti pəncərəyə keçməkdən əvvəl işləyəcəyimiz cədvəllər ilə bağlı məlumat verməliyəm.

1. Sales

Satılmış xidmət haqqında tam məlumatı göstərir. İstifadə olunacaq sütunlar:

* sub\_code - xidmətin unikal abunə kodu
* status - xidmətin statusu
* s\_price – xidmətə görə ödənilmiş məbləğ
* sub\_start – xidmətə abunənin başlanma tarixi
* sub\_end - xidmətə abunənin bitmə tarixi
* sub\_term – xidmətə abunə müddəti (gün ilə)
* pmt\_type – ödəniş yolu
* appeal\_type – müraciət yolu
* repr\_plc – müqavilənin imzanaldığı yer
* service\_id – Xidmətin kodu

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *SELECT TOP(5) sub\_code, s\_price, CAST(sub\_start as date) AS [start\_date], CAST(sub\_end as date) AS [end\_date], sub\_term, status, pmt\_type, appeal\_type, repr\_plc, service\_id*  *FROM Sales;* | | | | | | | | | |
| sub\_code | **s\_price** | **sub\_start** | **sub\_end** | **sub\_term** | **status** | **pmt\_type** | **appeal\_type** | **repr\_plc** | **service\_id** |
| 0583765 | 29.97 | 2016-03-31 | 2016-06-30 | 90 | END | CCD | ONL | OCS | HST |
| 0447981 | 30.00 | 2016-01-09 | 2016-07-08 | 180 | END | CCD | ONL | OCS | OTR |
| 1333962 | 131.64 | 2017-10-07 | 2018-10-06 | 365 | ACT | CCD | ONL | OCS | IAAS |
| 0007710 | 340.80 | 2013-09-18 | 2015-09-17 | 730 | END | CSH | PSN | PN1 | VPS |
| 1029724 | 3.97 | 2017-08-07 | 2017-08-07 | 1 | END | CCD | ONL | OCS | OTR |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1. Services   Xidmətlər lüğəti, xidmət kodu üzrə açıqlama verir.   * id – Xidmət kodu * name – Xidmət adı  |  |  | | --- | --- | | *SELECT id, name*  *FROM Services;* | | | Id | **name** | | HST | Xosting xidmətləri | | VPS | Virtual Şəxsi Xosting | | IAAS | İnfrastruktur Xidmət Kimi | | OTR | Digər | | 1. Payment\_mean   Ödəniş növü lüğəti, ödəniş tipi kodu üzrə açıqlama verir.   * id – tipin kodu * name – tipin açıqlaması  |  |  | | --- | --- | | *SELECT id, name*  *FROM payment\_mean;* | | | Id | **name** | | CSH | Nağd pul | | CCS | Bank kartı | |
| 1. Appeal\_type   Müraciət növü lüğəti. Müraciət kodu üzrə açıqlama verir.   * id – tipin kodu * name – tipin açıqlaması  |  |  | | --- | --- | | *SELECT id, name*  *FROM payment\_mean;* | | | Id | **name** | | ONL | Sayt / Online müraciət | | PSN | Təmsilçilik üzərindən müraciət | |  |

Nəzərə alsaq ki yaradılacaq hesabat agregassiya olunacaq, yəni vizual olaraq ilkin məlumat verəcək və 10 nəfərə yaxın işçi tərəfindən gündə azı 80 dəfə istifadə olunacaq(cəmi) o zaman yaradacağımız sorğu da maksimal olaraq çevik və yüngül olmalıdır. Bu səbəbdən ilk sorğumuzda SQL-in “agregate” funksiyalarından istifadə edəcəyik.

---------Ümumi satış haqqında məlumat ----------------

COUNT(s.status) as [Cəmi say ],

SUM(s.s\_price) as [Cəmi məbləğ],

MAX(s.s\_price) as [Maksimal xidmət alış qiyməti],

MIN(s.s\_price) as [Minimal xidmət alış qiyməti],

AVG(s.sub\_term) as [Orta xidmət müddəti],

AVG(s.s\_price) as [Orta xidmət alış qiyməti],

(SELECT COUNT(s.status) FROM sales s WHERE (sub\_start between @startdate and @enddate) and s.status = ‘CNS’) as [Alişdan sonra imtina],

---------Satış və ödəmə yolları üzrə məlumat -----

(SELECT COUNT(s.status) FROM sales s WHERE (sub\_start between @startdate and @enddate) and s.status in (‘ACT’,’END’) and s.pmt\_type\_id = 'CSH') as [Nagd emeliyyat sayi],

(SELECT COUNT(s.status) FROM sales s WHERE (sub\_start between @startdate and @enddate) and s.status in (‘ACT’,’END’) and s.pmt\_type\_id = 'CCD') as [Bank karti ile emeliyyat sayi],

(SELECT SUM(s.s\_price) FROM sales s WHERE (sub\_start between @startdate and @enddate) and s.status in (‘ACT’,’END’) and s.pmt\_type\_id = ‘CSH’) as [Nagd emeliyyat SUM],

(SELECT SUM(s.s\_price) FROM sales s WHERE (sub\_start between @startdate and @enddate) and s.status in (‘ACT’,’END’) and s.pmt\_type\_id = ‘CCD’) as [Bank karti ile emeliyyat SUM],

(SELECT COUNT(s.status) FROM sales s WHERE (sub\_start between @startdate and @enddate) and s.status in (‘ACT’,’END’) and s.appeal\_type\_id = 'ONL') as [Onlayn alish sayi],

(SELECT COUNT(s.status) FROM sales s WHERE (sub\_start between @startdate and @enddate) and s.status in (‘ACT’,’END’) and s.appeal\_type\_id = ‘PSN’) as [Filial alish sayi],

(SELECT SUM(s.s\_price) FROM sales s WHERE (sub\_start between @startdate and @enddate) and s.status in (‘ACT’,’END’) and s.appeal\_type\_id = 'ONL') as [Onlayn alish SUM],

(SELECT SUM(s.s\_price) FROM sales s WHERE (sub\_start between @startdate and @enddate) and s.status in (‘ACT’,’END’) and s.appeal\_type\_id = ‘PSN’) as [Filial alish SUM],

-----Xidmətlər üzrə məlumat ----

-------Xosting xədməti

(SELECT COUNT(s.status) FROM sales s WHERE (sub\_start between @startdate and @enddate) and s.status in (‘ACT’,’END’) and s.service\_id= ‘HST’) as [Count by HST],

(SELECT SUM(s.s\_price) FROM sales s WHERE (sub\_start between @startdate and @enddate) and s.status in (‘ACT’,’END’) and s.service\_id= ‘HST’) as [Sum by HST],

(SELECT AVG(s.s\_price) FROM sales s WHERE (sub\_start between @startdate and @enddate) and s.status in (‘ACT’,’END’) and s.service\_id= ‘HST’) as [AVG Sum by HST],

(SELECT AVG(s.sub\_term) FROM sales s WHERE (sub\_start between @startdate and @enddate) and s.status in (‘ACT’,’END’) and s.service\_id= ‘HST’) as [AVG Term by HST],

-------Virtual Şəxsi Server

(SELECT COUNT(s.status) FROM sales s WHERE (sub\_start between @startdate and @enddate) and s.status in (‘ACT’,’END’) and s.service\_id= ‘VPS’) as [Count by VPS],

(SELECT SUM(s.s\_price) FROM sales s WHERE (sub\_start between @startdate and @enddate) and s.status in (‘ACT’,’END’) and s.service\_id= ‘VPS’) as [Sum by VPS],

(SELECT AVG(s.s\_price) FROM sales s WHERE (sub\_start between @startdate and @enddate) and s.status in (‘ACT’,’END’) and s.service\_id= ‘VPS’) as [AVG Sum by VPS],

(SELECT AVG(s.sub\_term) FROM sales s WHERE (sub\_start between @startdate and @enddate) and s.status in (‘ACT’,’END’) and s.service\_id= ‘VPS’) as [AVG Term by VPS],

------- Infrastruktur Xidmət Kimi

(SELECT COUNT(s.status) FROM sales s WHERE (sub\_start between @startdate and @enddate) and s.status in (‘ACT’,’END’) and s.service\_id= ‘IAAS’) as [Count by IAAS],

(SELECT SUM(s.s\_price) FROM sales s WHERE (sub\_start between @startdate and @enddate) and s.status in (‘ACT’,’END’) and s.service\_id= ‘IAAS’) as [Sum by IAAS],

(SELECT AVG(s.s\_price) FROM sales s WHERE (sub\_start between @startdate and @enddate) and s.status in (‘ACT’,’END’) and s.service\_id= ‘IAAS’) as [AVG Sum by IAAS],

(SELECT AVG(s.sub\_term) FROM sales s WHERE (sub\_start between @startdate and @enddate) and s.status in (‘ACT’,’END’) and s.service\_id= ‘IAAS’) as [AVG Term by IAAS],

-------Digər

(SELECT COUNT(s.status) FROM sales s WHERE (sub\_start between @startdate and @enddate) and s.status in (‘ACT’,’END’) and s.service\_id= ‘OTR’) as [Count by OTR],

(SELECT SUM(s.s\_price) FROM sales s WHERE (sub\_start between @startdate and @enddate) and s.status in (‘ACT’,’END’) and s.service\_id= ‘OTR’) as [Sum by OTR],

(SELECT AVG(s.s\_price) FROM sales s WHERE (sub\_start between @startdate and @enddate) and s.status in (‘ACT’,’END’) and s.service\_id= ‘OTR’) as [AVG Sum by OTR],

(SELECT AVG(s.sub\_term) FROM sales s WHERE (sub\_start between @startdate and @enddate) and s.status in (‘ACT’,’END’) and s.service\_id= ‘OTR’) as [AVG Term by OTR]

FROM

sales s

WHERE

sub\_start between @startdate and @enddate

and s.status in (‘ACT’,’END’)

Qeyd: Sorğunda qeyd olunmuş *@startdate* və *@enddate* **parametrlərdir,** biz onları növbəti addımlarda sazlayacayıq.

Gördüyünüz kimi sorğuda çoxlu daxili sorğulardan istifadə olunur. Bu bizə əsas qaydalardan birinə riayət etməyə imkan verdi – “əgər məlumat üzərində hər hansı bir Məlumat Bazası Serveri tərəfindən daha effektiv görülə biləcək iş aparmaq lazımdırsa o zaman bu iş mütləq MBS tərəfindən görülməlidir”. Bizim halda seçilmiş tarix aralığından aslı olaraq yüz minlərlə sətr üzərində işi daha effektiv bunun üçün nəzərdə tutulmuş SQL Server görəcək.

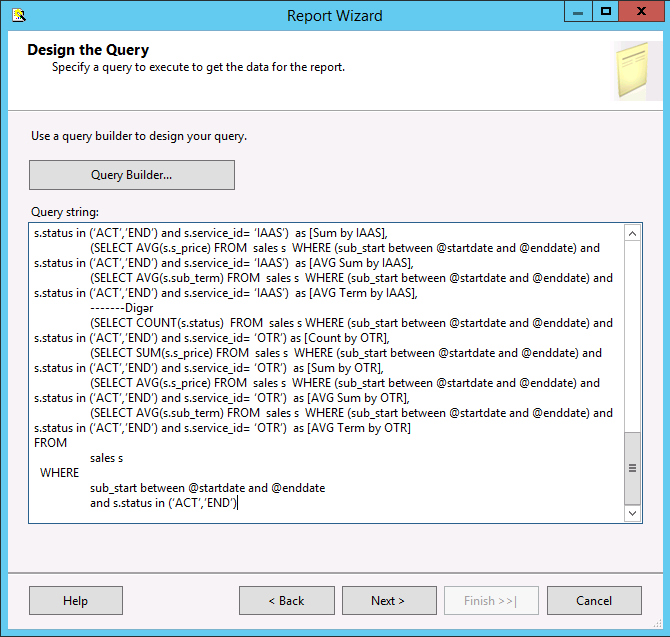
Əlbəttə ki sorğunu daha qısa yazmaq, xidmətlər üzrə filtrlərin çox bir halda eyni olduğunu nəzərə alaraq gruplaşdırmaq olar, amma bu tutorialda sorğunun oxunmasını, başa düşməni və “debugging”-I asanlaşdırmaq üçün bu yoldan istifadə olunur.

Sorğunu optimallaşdırmaq və debugging asanlaşdırmaq yollarından biri biri də 4 ayrı-ayrı DATASET yaratmaq idi:

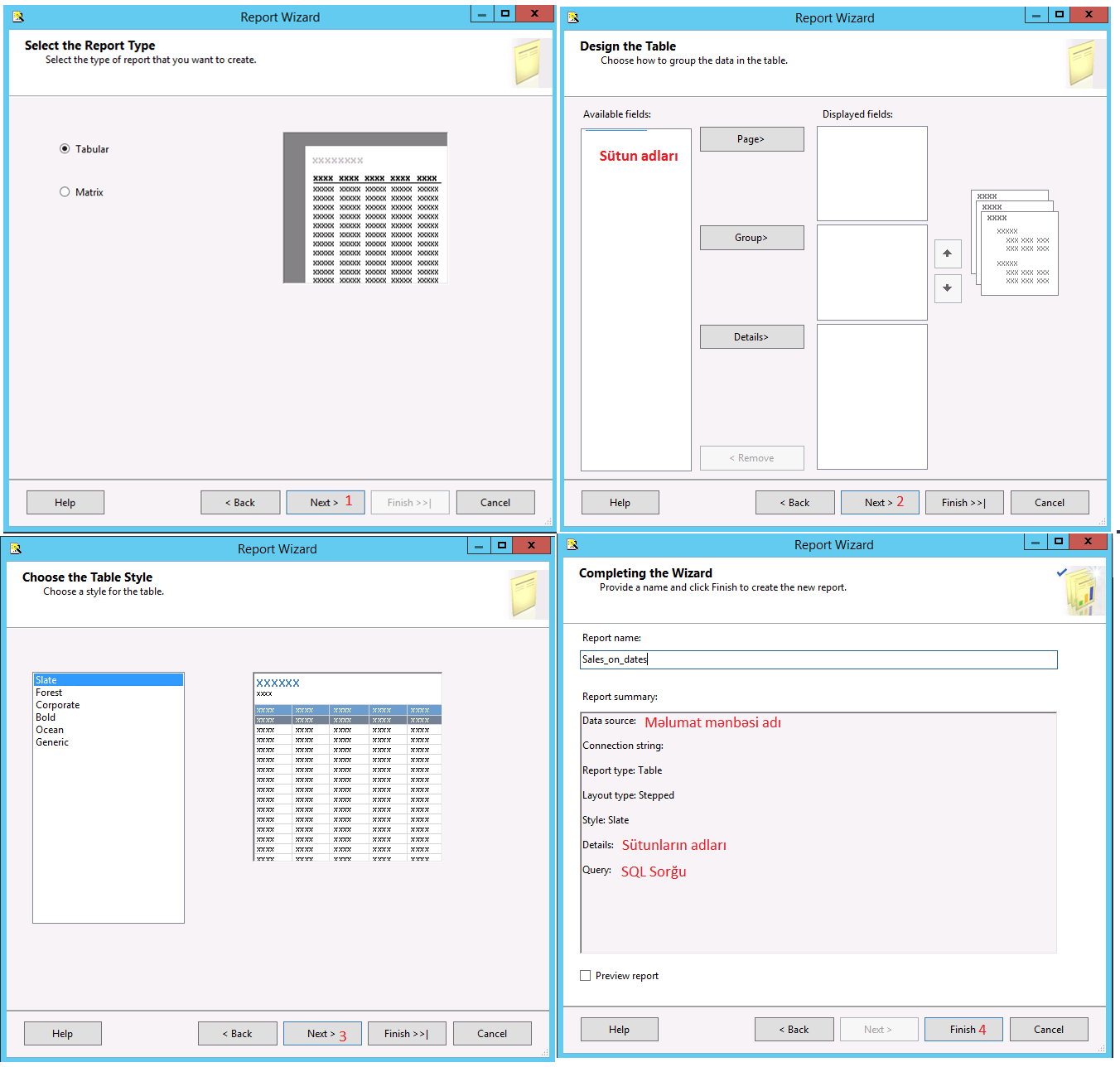
1. Satış haqqında ümumi məlumat – 7 sütun və 1 sıra.
2. Ödəmə yolları haqqında məlumat – 2 sütun, 2 sıra
3. Müraciət yolları haqqında məlumat – 2 sütun, 2 sıra
4. Xidmətlər haqqında məlumat – 4 sütun və 4 sıra

Beləliklə hazırladığımız sorğu 31 sütun və 1 sətrdən ibarətdir. 31 sütun böyük rəqəm görsənsə də hesabat formasın sazlayan zaman məlumatın 1 sıra olmasından və bununla hər bir pozisiya üzrə məlumatın 1 sütun və 1 sıra üzrə kəsişdiyindən hesabatdaxili filtrlərdən ümumiyyətlə istifadə etməyəcəyimizə gətirəcək ki bu da hesabatın formalaşmasına və hesabat üzrərində işi sadə formatlaşmaya gətirəcək.

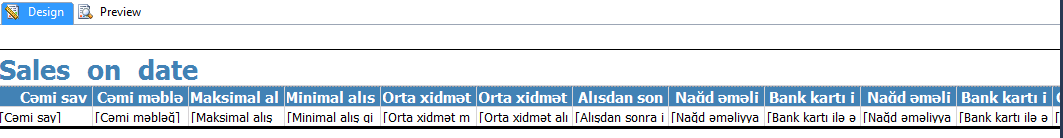
Sorğunu hazır etdikdən sonra pəncərəyə kopyalıyıb “**NEXT >**” düyməsinə basırıq.



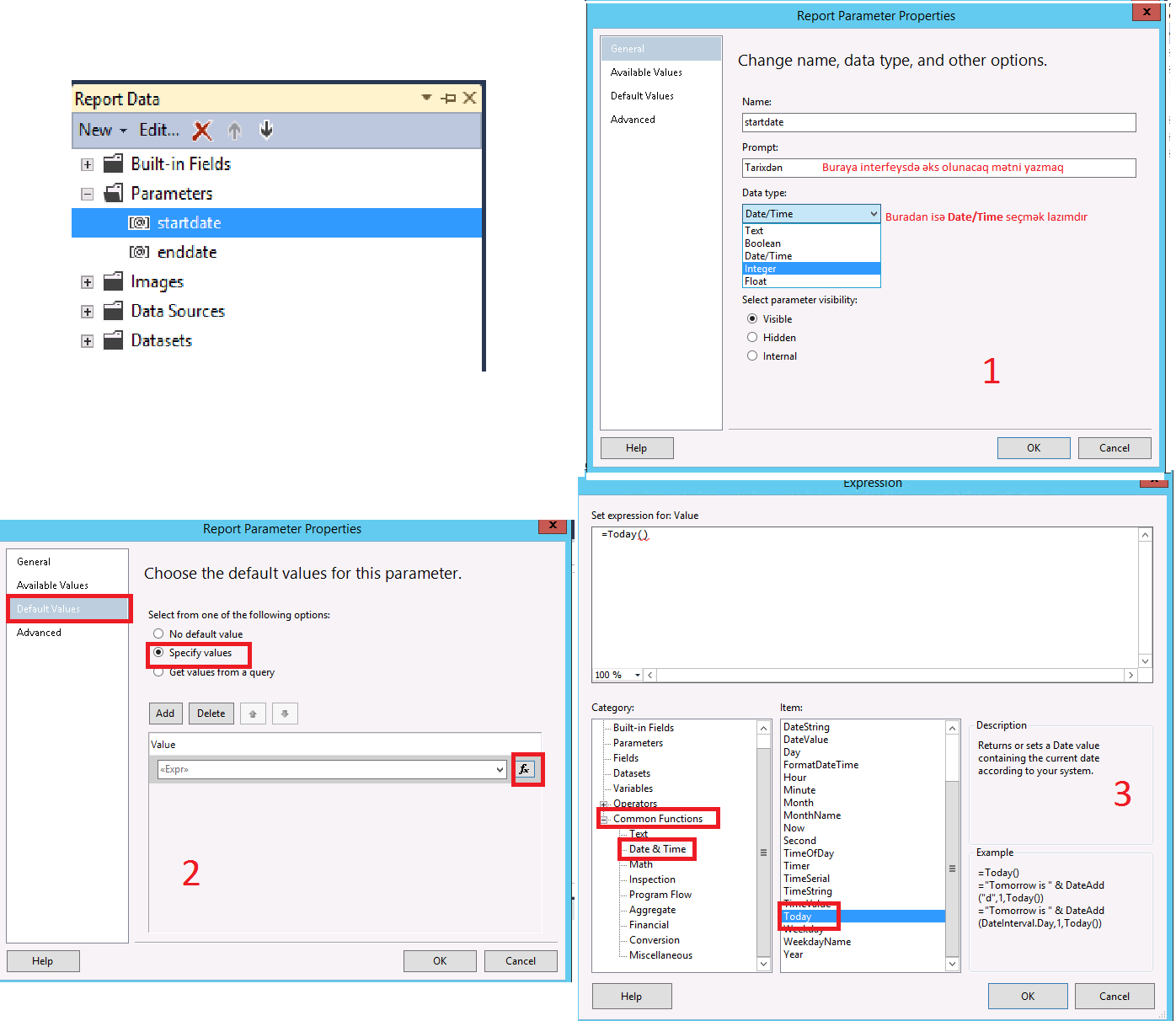
Bir daha “**NEXT >**”



“**Finish**”-ə basdıqdan sonra ilkin hesabat forması hazır olur



Hesabat ilə tanış olmaq üçün “Preview” –a tıklamaq lazımdır, amma hələ ki hesabat işləməyəcək. İşləməsi üçün sorğiuda qeyd olunmuş *parameterlərin* tipini və standart dəyərlərini qeyd etməliyik. Onları sazlamaq üçün ekranın sol tərəfində yerləşən “*Report Data”* panelinə daxil olub *Parameters* qovluğunda parameter üzərinə tıklamaq lazımdır.

Gördüyünüz kimi standart dəyərlər “Bu gün” olaraq seçilmişdir, və standart dəyəri qeyd etməklə biz iki dovşanı bir güllə ilə vura bildik. Həm hesabat açılarkən nəzərdə tutulmuş funksiyanı (gündəlik satışın əks olunması) icra edir, həm də hesabat yaradıması zamanı “yaxşı praktika” sayılan dəyərləri qeyd etdik.

Beləliklə artıq hesabat işləkdir. Biz “Preview”-ə daxil olub oradan hesabatın düzgün əks olunduğuna əmin olduğundan sonra hesabatın içərisində olan cədvəli işarələyib .... “*Delete*” düyməsi ilə pozmaq olar. Yuxarıda qeyd etdiyim kimi mən hesabatı “Köməkçidən” istifadə etmədən işləməyi tərciə edirəm, həm də etdiyimiz hesabat intuitiv olmalıdır, amma 31 sütunluq hesabatı “oxumaq” çətin ola bilər.

Pozumuş cədvəl əvəzinə məlumatı yuxarıda qeyd etdiyim məntiqi bloklar üzrə bölüb yerləşdirəcəyik.

İlk öncə hesabat başlığın üzərinə iki dəfə tıklayıb adın “Tarix aralığına satış” qoyaq. Ondan sonra boş sahə üzərinə sıçanın sağ düyməsinə basaraq **“Insert”🡪”Image”** seçib şəkil əlavə etmə pəncərəsinə daxil olaq. Pəncərənin ilk səhifəsində şəkil mənbəsin “*Embedded*” seçib “*İmport*” düyməsi vasitəsi ilə kompyuterdən yüngül logo seçib başlığın yanında yerləşdirək.

Aralığ nəticələrə baxaq:





Davam edə bilərik. Növbəti addım boş sahəyə məlumat əks olunması üçün istifadə olunan əsas alətlərdən olan cədvəlin əlavə olunmasıdır. Boş sahəyə sağ düymə🡪**Insert🡪Table**. Yaradılmış cədvəlin *Header* hissədə başlıq, *Data* hissəsində məlumatı yerləşəcək.

Növbəti addımlarda istifadə olunması üçün cədvəl şablonu yaratmaq lazımdır, amma ondan əvvəl görünüşü planlaşdırmaq lazımdır. Məlumat məntiq üzrə cəmi 2 bloka və 7 cədvələ bölünəcək:

Blok #1. Ümumi satış göstəriciləri.

* Ümumi satış haqqında məlumat ( 1 cədvəl: 7 sütun və 2 cərgə)
* Satış və ödəmə yolları üzrə məlumat ( 2 cədvəl: hər biri 4 sütun və 2 cərgə) (bu və aşağıda qeyd olunan hallarda “Table” əvəzinə “Matrix”-dən də istifadə etmək olardı, amma istifadə etdiyimiz sorğu artıq buna imkan vermir)

Blok #2. Xidmətlər üzrə məlumat.

* Hər satılan mal üzrə umumi göstərici ( 4 cədvəl: Hər biri 4 sütun və 2 cərgə)

İlk öncə ilk artıq yaratdığımız cədvəlin üzərinə sağ düymə basıb **“Insert Column”🡪”To Right”** ilə sütun sayın 7-yə çatdırırıq. Sonra *Header* hissəsinə Ümumi satış göstəriciləri üzrə başlıqları yazırıq.



Sonra kursoru *Data* cərgəsində olan müvafiq xananın yuxarı sağ küncünə yaxınaşdırıb şəkidə gördüyünüz vasitəçi üzərinə tiklayırsınız. Açılacaq keçiddə “*DataSet1”*-ə keçib müvafiq “*Cəmi say*” pozisiyasın seçmək lazımdır. Bu əməliyyat hər bir xana üzərində apararılmalıdır. İş bitdikdən sonra aralıq heabat şablonu bu cür olmalıdır.



**“Preview”** vasitəsi ilə baxaq:



Əla! Amma vizual görüntü ilə problemlər var:

* Rəqəmlər formatlaşmayıb.

Həll: “Data” cərgəsindəki xana üzərində sağ düymə “Text Box properties”🡪’Number’🡪’Number’

* Decimal places: 2
* Use ‘,’ as separator
* Show zero as: ‘-‘
* Bəzi pul ilə bağlı göstəricilər sadə rəqəm kimi göstərilib.

Həll: “Data” cərgəsindəki xana üzərində sağ düymə “Text Box properties”🡪’Number’🡪’Currency’

* Decimal places: 2
* Use ‘,’ as separator
* Show zero as: ‘-‘
* Symbol: ‘AZN’
* Show symbol after value
* Include a space
* Bəzi xanalar çox qısa, bəziləri çox uzundur.

Həll: Sadəcə sütun sərhədlərin maus ilə tənzimlənir.

* Başlıqlar adi məlumatdan vizual olaraq ayrılmır.

Həll: Başlıq xanaları üzərinə sağ düymə: “Text Box properties”🡪’Font’

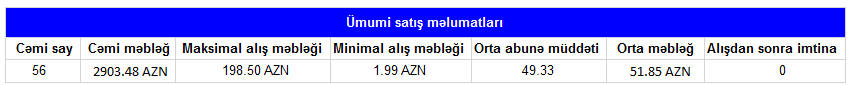
* + Bold
  + Font color

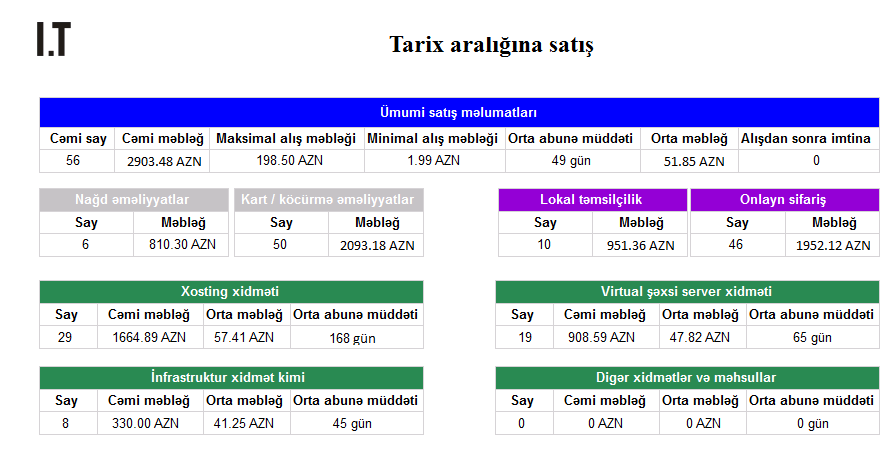
“Text Box properties”🡪’Fill’

* + Fill color
* Ayrı vizual çatışmamazlıqlar.

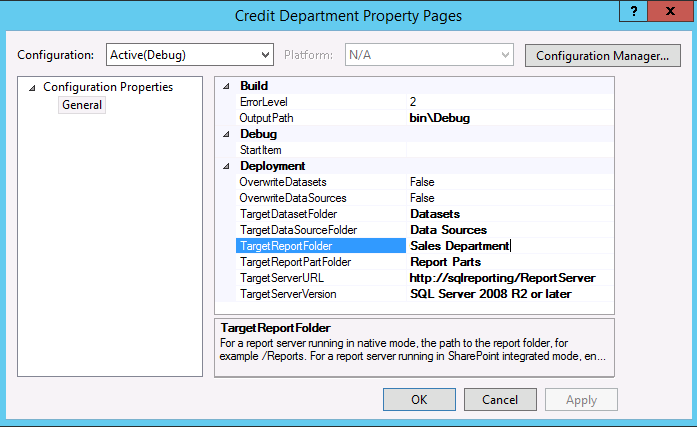
Həll: Həm “Data” həm “Header” xanaları üçün. Xana üzərinə sağ düymə: “Text Box properties”🡪’Alignment’

* + Horizontal: Middle
  + Vertical: Center

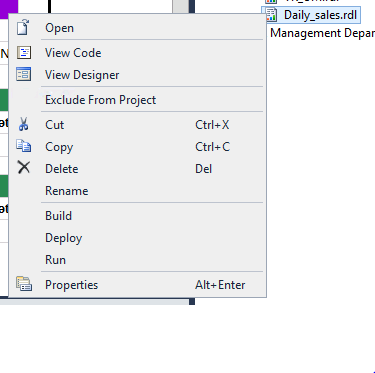
</details>

İndi isə qeyd etdiyim əməliyyatları qalan cədvəllər üzrə aparaq və hesabata yenidən baxaq.

Beləliklə hesabat hazırdır. Hesabatı serverdə dərc etmək üçün “İTSER” proyektinin ayarlarına daxil olub “TargetReportFolder” xanasının dəyərini Satiş Şöbəsinin SSRS serverində qovluğuna yönləndirmək lazımdır. Hesabat “TargetServerURL” addresi üzrə “TargetReportFolder” govluğunda dərc olunacaq.



Dərc etmək üçün hesabat üzərindən “Deploy” seçməlisiniz.



Beləliklə ilk məqalə bitdi. Gələn məqalələrdə hesabatlar arası parameterlərin ötürülməsi, “Hesabat yaratmağ zamanı 10 “best practices” bölüşəcəm.