

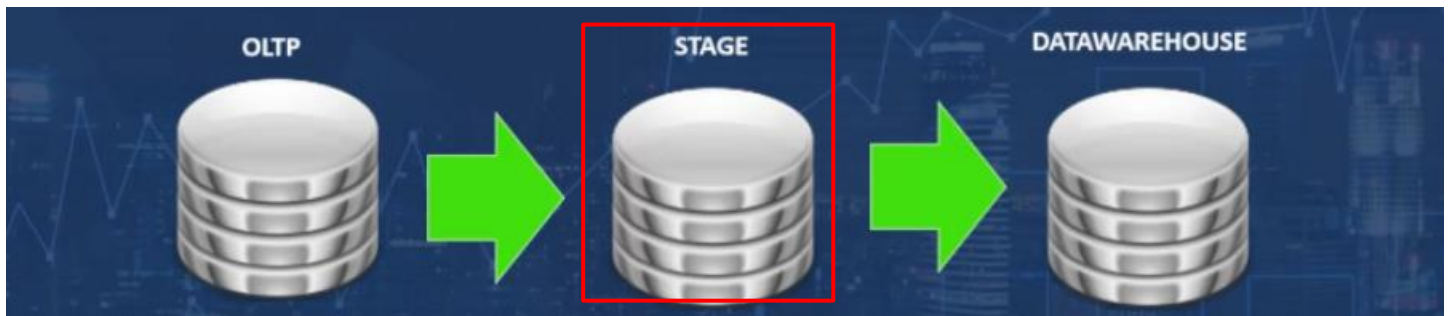
O ambiente de *Stage*, ou *Stage Area*, é utilizado como um intermediário entre o ambiente OLTP, e o *Data Warehouse*. É nesse ambiente em que se realiza a descarga dos dados brutos, aplicando alguma transformação necessária no momento da carga para que então seja transferida para o *Data Warehouse*.

As siglas OLTP e OLAP definem as fronteiras das funções dos bancos as quais elas são submetidas: OLTP ou *On-line Transaction Processing*, termo usado para referir os sistemas transacionais que são alimentados continuamente pelos sistemas de ponta da empresa com interface ao usuário. Um sistema de vendas por exemplo, com as informações de venda inseridas ali, alimenta o banco transacional OLTP.

Já a sigla OLAP, ou *On-line Analytical Processing*, define a área disponível para processamento e análise de dados provenientes do ambiente OLTP prévio. Nesse ponto, inicia-se a aplicação do *Business Intelligence*.

A área de *Stage* tem o objetivo de desafogar o OLTP e o *Data Warehouse* ao mesmo tempo, pois é aqui que os dados são espelhados e trabalhados durante a carga, liberando o processo de trabalho do banco transacional e do DW, evitando lentidões, uma vez que é uma boa prática não conectar diretamente ao ambiente OLTP para realizar queries durante a rotina de trabalho.

Nesse momento, trabalharemos na segunda etapa do projeto: Stage Area



A *Stage Area*, é construída de acordo com os requisitos do *Data Warehouse*, que por sua vez é construído de acordo com o plano de gestão de empresa, através do levantamento de requisitos junto aos gestores do negócio.

O levantamento de requisitos é basicamente o *core* do projeto de *Business Intelligence*, pois aí serão armazenadas as informações em formato necessário para extração de valor e insights a partir de dados históricos.

Os requisitos levantados para esse projeto em que o *Data Warehouse* objetiva responder, são:

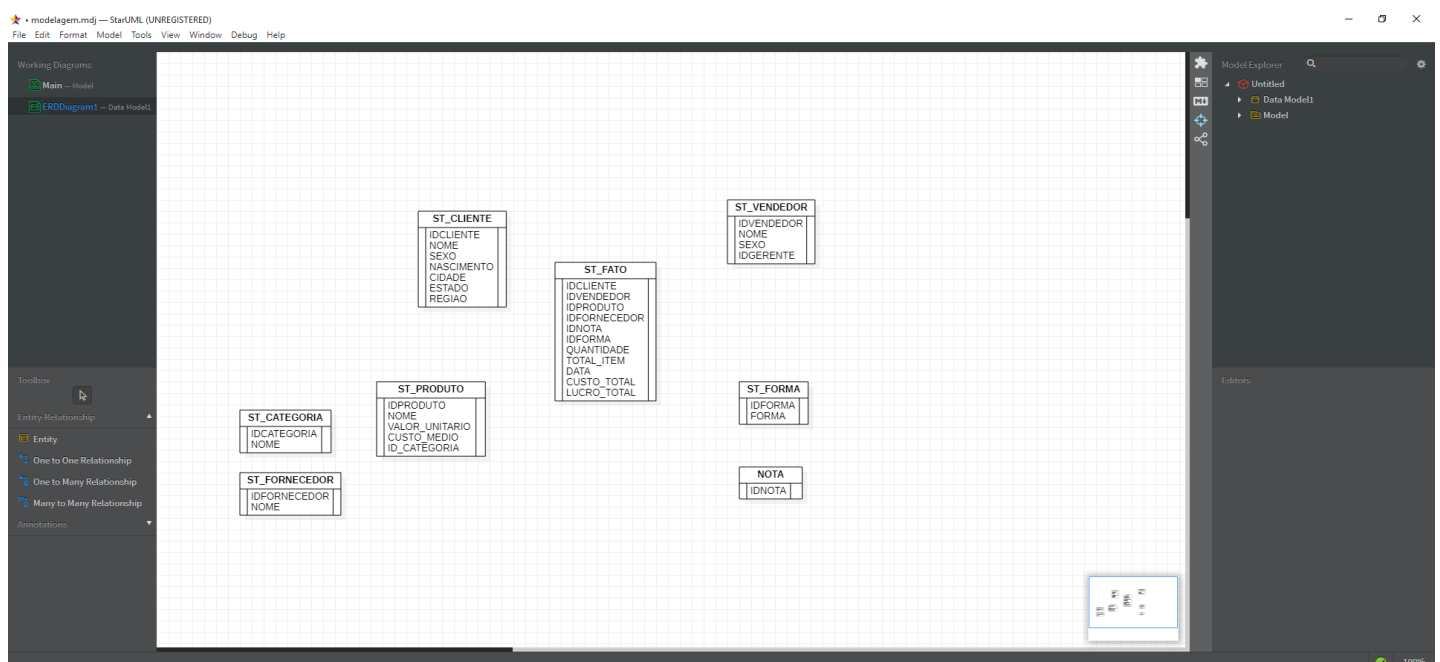
- Quem são os melhores clientes?
- Quem são os melhores vendedores?
- Qual categoria rende mais?
- Qual a minha relação com os fornecedores?
- Qual meu pior e melhor produto?
- Em qual região eu vendo mais?

Essas perguntas serão cruciais para a modelagem do DW, pois é lá que as consultas serão realizadas para a gestão do negócio. Importante ressaltar que a *Stage Area* é modelada de acordo com os requisitos do *Data Warehouse*, garantindo o fluxo contínuo de cargas de dados.

MODELAGEM LÓGICA DA STAGE AREA (COMERCIO_STAGE)

Com o escopo do projeto em mãos, inicia-se o processo de *modelagem lógica* da área de Stage para as cargas de dados.

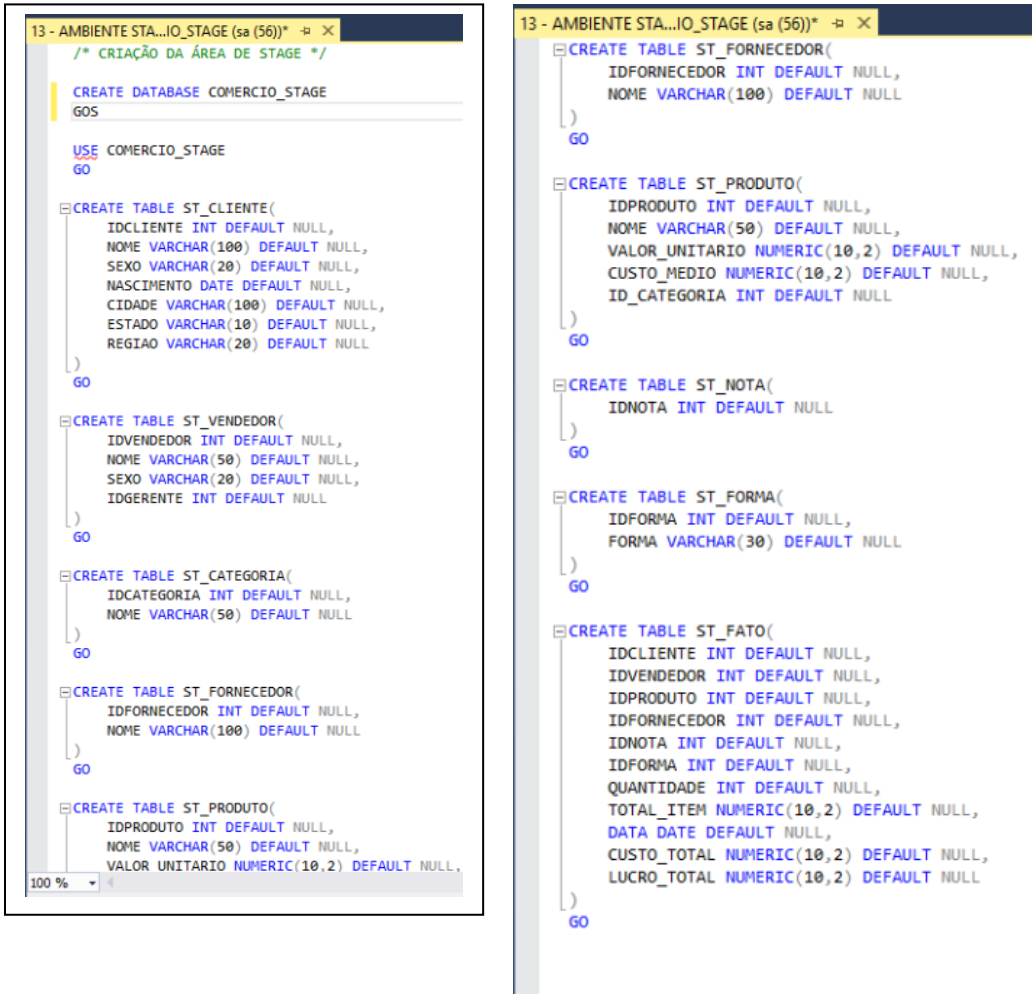
O aplicativo utilizado para a modelagem de dados, foi o **StarUML**, em sua versão gratuita. O aplicativo oferece ótimo desempenho e uma visão geral de como serão definidos as tabelas e os relacionamentos, quando aplicável.



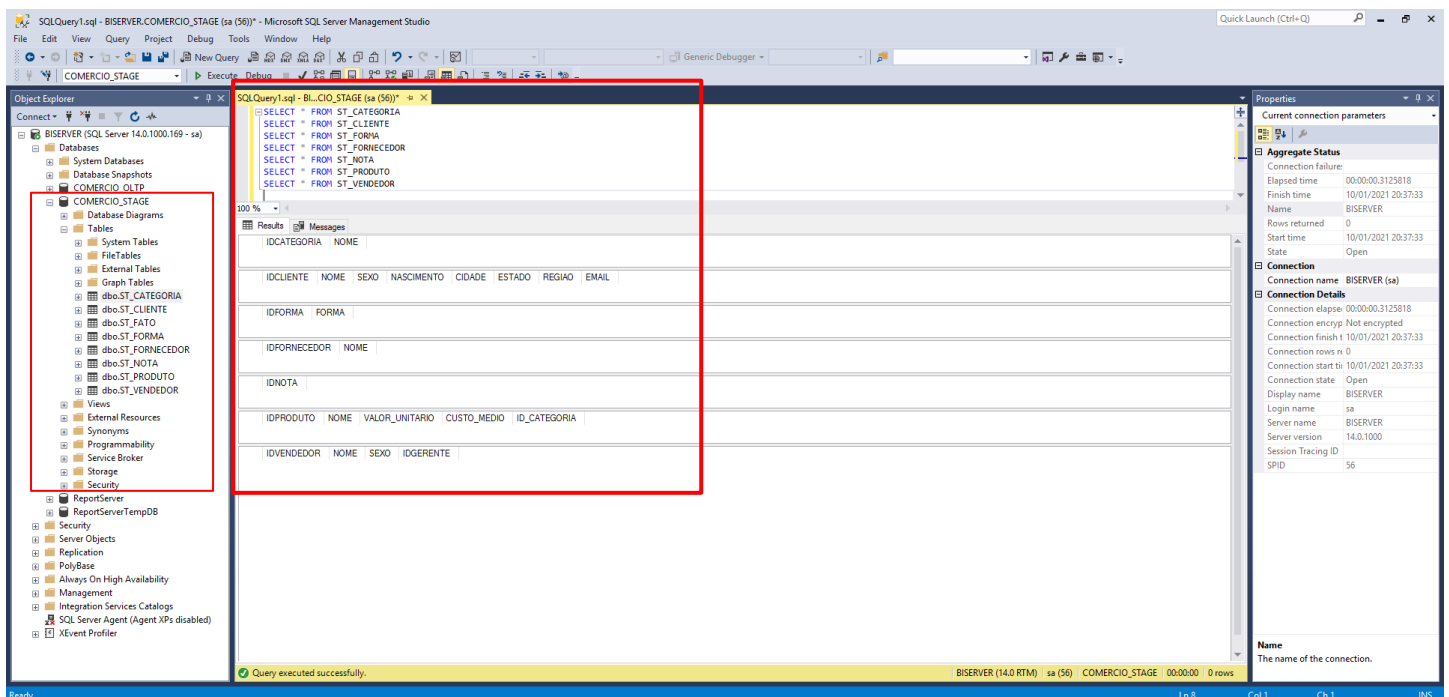
Na área de *Stage*, é importante salientar que os relacionamentos e *constraints* não são definidos pois não há – aqui – uma preocupação com a integridade relacional, uma vez que os dados foram transferidos diretamente das relações do ambiente OLTP. Essa área contém os dados que serão transferidas para o DW, que então serão construídas todas as relações e *constraints* para garantir a granularidade e a integridade relacional.

MODELAGEM FÍSICA DA STAGE AREA (COMERCIO_STAGE)

Inicia-se o processo de modelagem física no banco de dados, utilizando o SQL Server 2017.



Nessa etapa, a modelagem da tabela ST_FATO será de grande importância para se ter uma visão geral dos dados de vendas. Essa será a principal tabela da *Stage*, onde já será possível extrair informações úteis. Como resultado das tabelas criadas no banco, pode-se agora começar o processo de carga de dados.



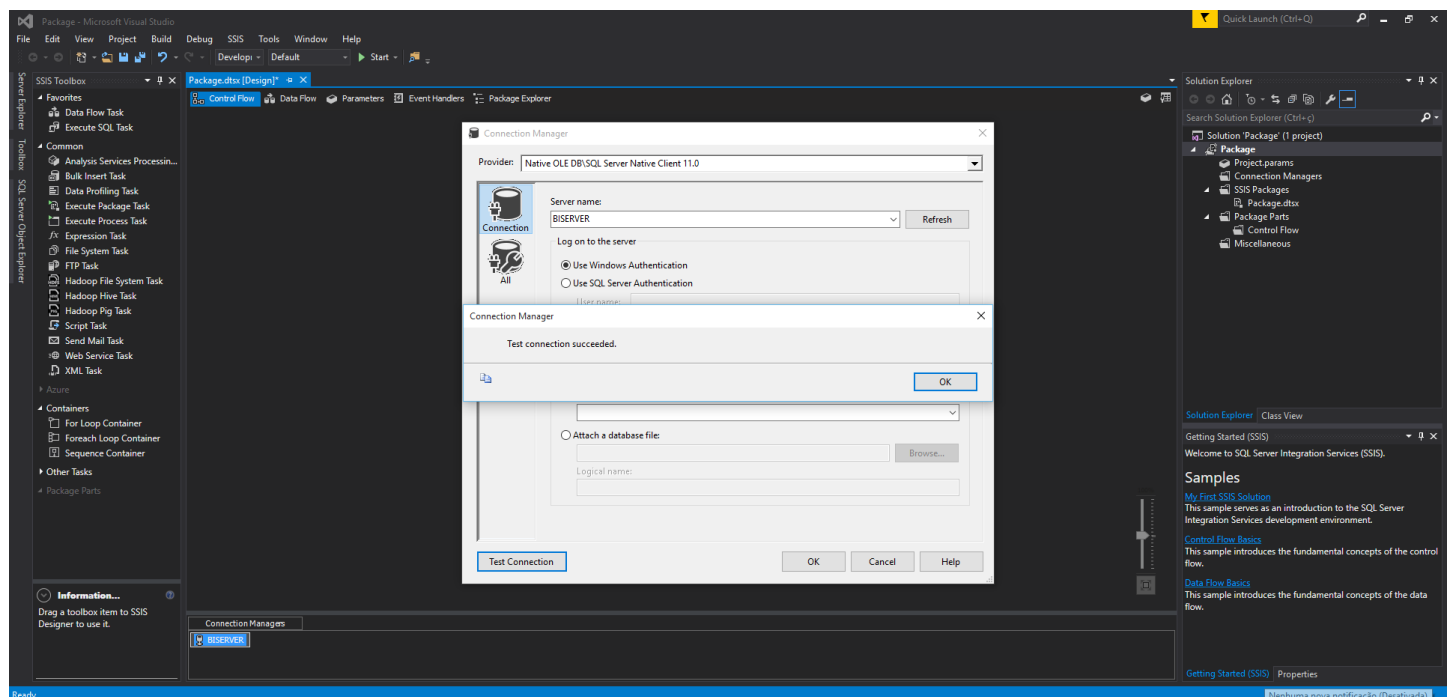
MICROSOFT SQL SERVER INTEGRATION SERVICES (SSIS)

O módulo Integration Services, ou SSIS, tem a função de criar pacotes de ETL (*extract, transform, load*), ou extração, transformação e carga.

Estes pacotes serão os responsáveis por criar funções automatizadas em linguagem SQL para interagir com o banco de dados OLTP, selecionando, extraído e carregando na área de *Stage*.

A suíte apresenta uma alta gama de funções para manipulação de dados, facilitando as transformações no momento da extração, despejando os dados já prontos para serem trabalhados e/ou consultados na área de *Stage*.

Primeiro, criamos a conexão com o servidor central e com os bancos disponíveis, *OLTP* e *STAGE*, e testamos a conexão:



ETL STAGE AREA – CARGA DE TABELAS

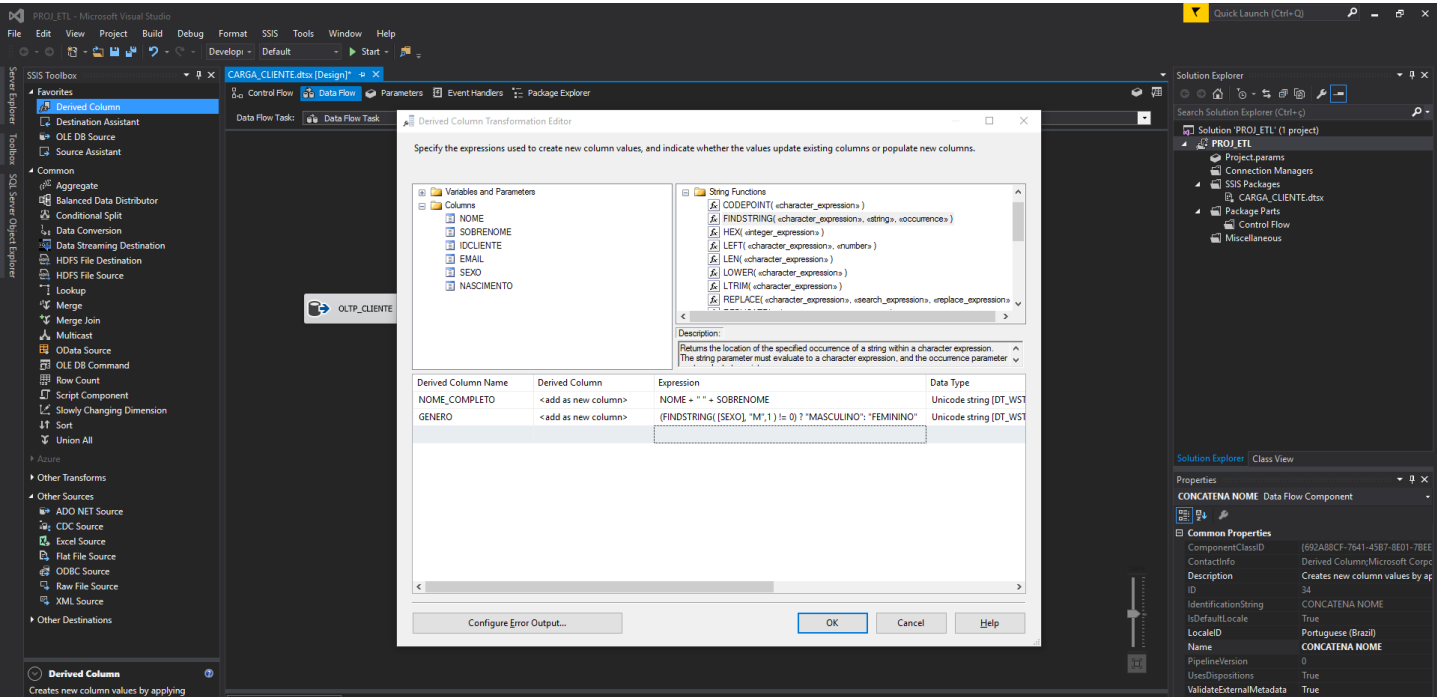
CARGA ST_CLIENTE

Para exemplificar uma transformação de dados no momento do *ETL* feito pelo SSIS, aplicaremos uma regra na carga da tabela ST_CLIENTE, na coluna “NOME” e na coluna “SEXO”.

Na coluna “NOME”, o processo de ETL irá concatenar as colunas NOME e SOBRENOME do banco OLTP em apenas uma coluna do banco STAGE, fazendo com que a tabela NOME se torne o nome completo do cliente, facilitando assim as *queries* realizadas.

Ao mesmo tempo, será criada uma regra para definir o gênero do cliente, onde o processo de ETL transforma os caracteres “M” e “F” em valores *booleanos*, que por sua vez são transformados em uma *string* “Masculino” e “Feminino”.

Dessa forma, as regras serão aplicadas no momento do carregamento e a área de *Stage* será carregada com os dados já transformados.

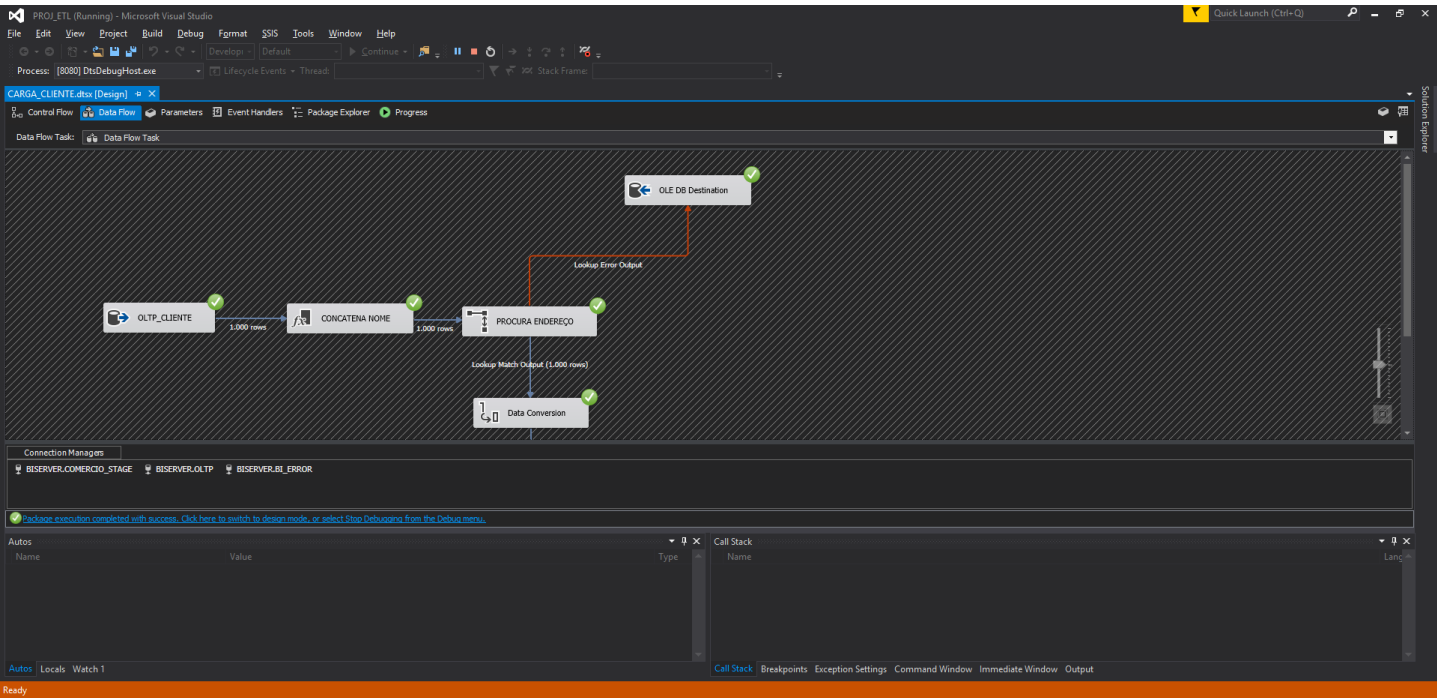


START ETL – Processo de carga dos dados

Aqui, os dados são carregados, com destino final à *Stage Area*.

A função do pacote apontando para a parte superior, indica um *flush* de dados que podem conter erros. Esses dados serão destinados à um banco de dados específico para tratamento de erros, chamado BI_ERROR, que então poderá ser consultada e realizado os devidos tratamentos para uma carga posterior.

Isso evita que o processo de ETL seja interrompido durante a carga de dados ao encontrar alguma linha com erro. Os erros poderão ser consultados e tratados posteriormente.



Os dados então são carregados e já podem ser consultados, utilizando um **SELECT**.

SQLQuery1.sql - BI...CIO_STAGE (sa (58))*

SELECT * FROM ST_CLIENTE

100 %

Results Messages

	IDCLIENTE	NOME	SEXO	NASCIMENTO	CIDADE	ESTADO	REGIAO	EMAIL
1	1	Luan Costa	MASCULINO	1981-04-22	Abreu e Lima	PE	NORDESTE	LuanRodriguesCosta@supemail.com
2	2	Eduardo Cunha	MASCULINO	1968-05-17	Abreu e Lima	PE	NORDESTE	EduardoCostaCunha@thyta.com
3	3	Thiago Castro	MASCULINO	1979-11-30	Abreu e Lima	PE	NORDESTE	ThiagoGomesCastro@supemto.com
4	4	Clara Souza	FEMININO	1989-11-07	Alagoinhas	BA	NORDESTE	ClaraLimaSouza@supemail.com
5	5	Aline Alves	FEMININO	1970-11-07	Alegrete	RS	SUL	AlineBarbosaAlves@cuvov.de
6	6	Caio Barbosa	MASCULINO	1990-05-11	Almirante Tamandaré	PR	SUL	CaioGomesBarbosa@amyspy.com
7	7	Kauã Cunha	MASCULINO	1953-09-21	Alvorada	RS	SUL	KausPintoCunha@supemail.com
8	8	Rafaela Gomes	FEMININO	1988-01-16	Americana	SP	SUDESTE	RafaelaRochaGomes@einrot.com
9	9	Samuel Lima	MASCULINO	1960-04-06	Americana	SP	SUDESTE	SamuelRibeiroLima@jourrapide.com
10	10	Kaua Rocha	MASCULINO	1952-10-23	Amparo	SP	SUDESTE	KauaMartinsRocha@fleckens.hu
11	11	Júlia Azevedo	FEMININO	1981-08-27	Amparo	SP	SUDESTE	JuliaDiasAzevedo@supemail.com
12	12	Murilo Silva	MASCULINO	1973-01-04	Ananindeua	PA	NORTE	MuriloPereiraSilva@jourrapide.com
13	13	Júlio Cunha	MASCULINO	1994-04-01	Ananindeua	PA	NORTE	JulioSilvaCunha@supemail.com
14	14	Amanda Melo	FEMININO	1950-02-19	Ananindeua	PA	NORTE	AmandaPintoMelo@amyspy.com
15	15	Tomás Oliveira	MASCULINO	1958-10-12	Anápolis	GO	CENTRO-OESTE	TomasSousaOliveira@gustr.com
16	16	Vinícius Barros	MASCULINO	1980-07-19	Anápolis	GO	CENTRO-OESTE	ViniciusCardosoBarros@jourrapide.com
17	17	Eduarda Gonçalves	FEMININO	1957-01-27	Aparecida de Goiânia	GO	CENTRO-OESTE	EduardaCavalcantiGoncalves@cuvov.de
18	18	José Pereira	MASCULINO	1968-01-18	Aparecida de Goiânia	GO	CENTRO-OESTE	JoseRochaPereira@einrot.com
19	19	Diogo Castro	MASCULINO	1961-03-20	Aparecida de Goiânia	GO	CENTRO-OESTE	DiogoAraujoCastro@gustr.com
20	20	Leonardo Almeida	MASCULINO	1953-08-10	Apucarana	PR	SUL	LeonardoAlvesAlmeida@amyspy.com
21	21	Eduarda Ferreira	FEMININO	1983-09-15	Apucarana	PR	SUL	EduardaBarosFerreira@supemto.com
22	22	Fábio Barbosa	MASCULINO	1984-11-02	Apucarana	PR	SUL	FabioMartinsBarbosa@thyta.com
23	23	Thais Fernandes	FEMININO	1988-09-11	Araçatuba	SP	SUDESTE	ThaisSantosFernandes@supemail.com
24	24	Kai Castro	MASCULINO	1959-05-21	Araçatuba	SP	SUDESTE	KaiGoncalvesCastro@thyta.com
25	25	Rodrigo Santos	MASCULINO	1956-07-03	Araçatuba	SP	SUDESTE	RodrigoSousaSantos@gustr.com
26	26	Rafael Sousa	MASCULINO	1959-03-14	Araçatuba	SP	SUDESTE	RafaelCostaSousa@cuvov.de
27	27	Kauã Pinto	MASCULINO	1952-12-09	Araguaína	TO	NORTE	KauaRodriguesPinto@amyspy.com
28	28	Gabriela Costa	FEMININO	1989-07-23	Araguaína	TO	NORTE	GabrielaMeloCosta@supemto.com
29	29	Carlos Ferreira	MASCULINO	1985-09-17	Araguari	MG	SUDESTE	CarlosPintoFerreira@jourrapide.com
30	30	Leila Melo	FEMININO	1955-06-03	Araguari	MG	SUDESTE	LeilaCastroMelo@supemail.com
31	31	Manuela Costa	FEMININO	1950-03-02	Arapiraca	AL	NORDESTE	ManuelaCorreiaCosta@jourrapide.com
32	32	Sofia Dias	FEMININO	1983-10-08	Araraquara	SP	SUDESTE	SofiaRochaDias@teleworm.us
33	33	Arthur Rodrigues	MASCULINO	1985-05-09	Araraquara	SP	SUDESTE	ArthurCastroRodrigues@nistr.com

Query executed successfully. | BISERVER (14.0 RTM) | sa (58) | COMERCIO_STAGE | 00:00:00 | 1000 rows

Criamos então os pacotes e realizamos o processo de *ETL* para as demais tabelas do banco *OLTP* para a *STAGE*.

Tabela ST_CATEGORIA

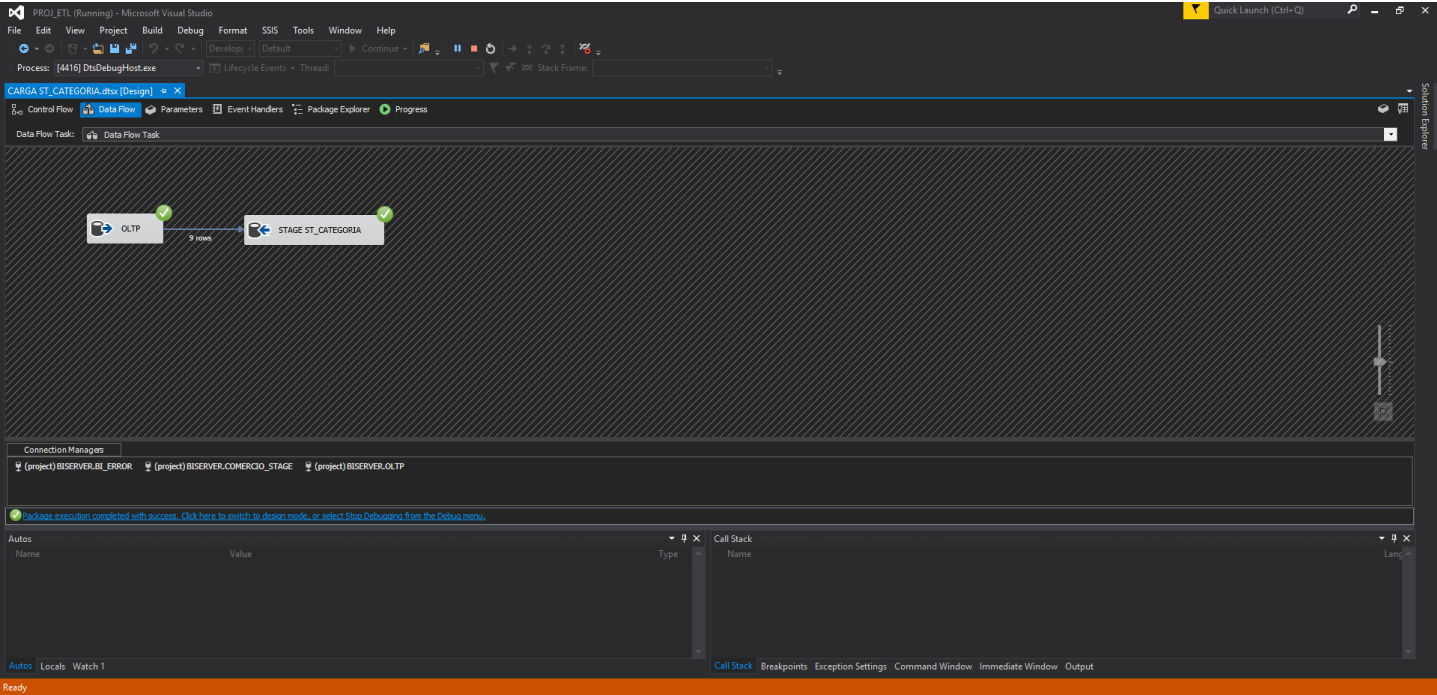


Tabela ST_VENDEDOR

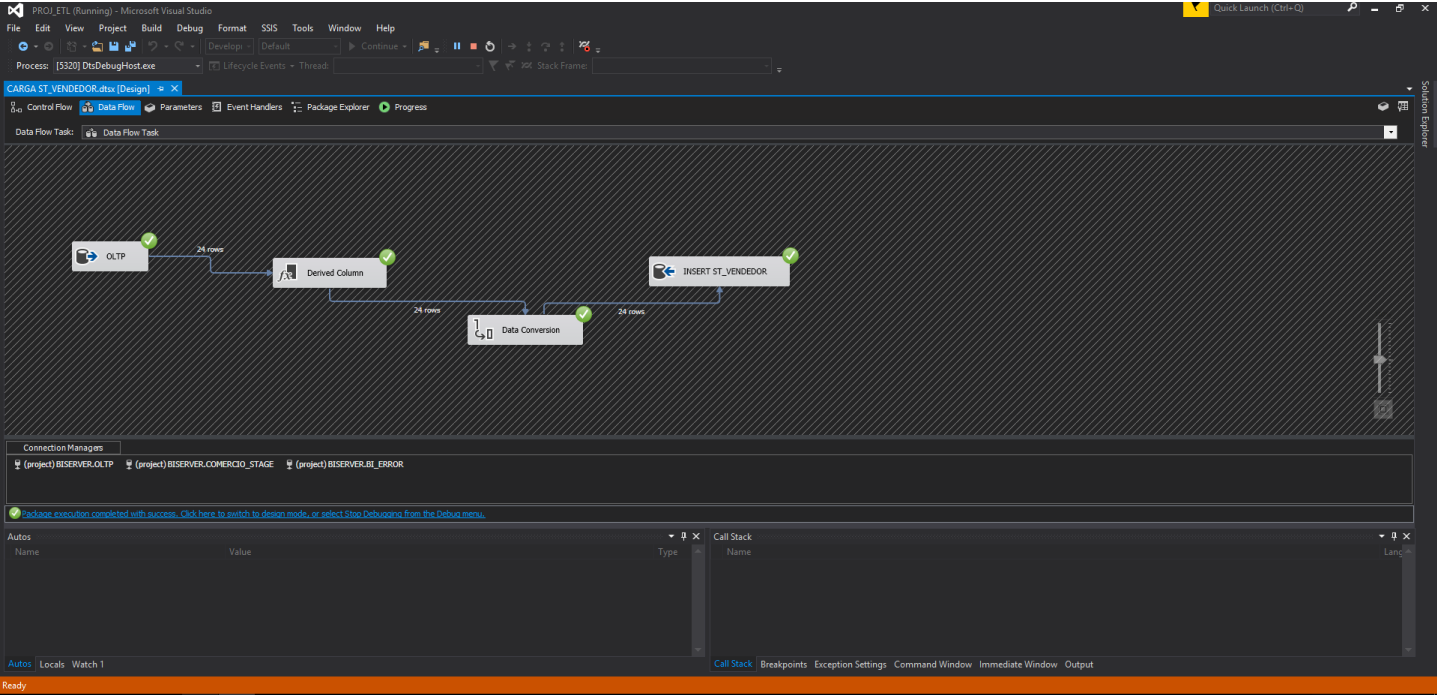


Tabela ST_FORNECEDOR

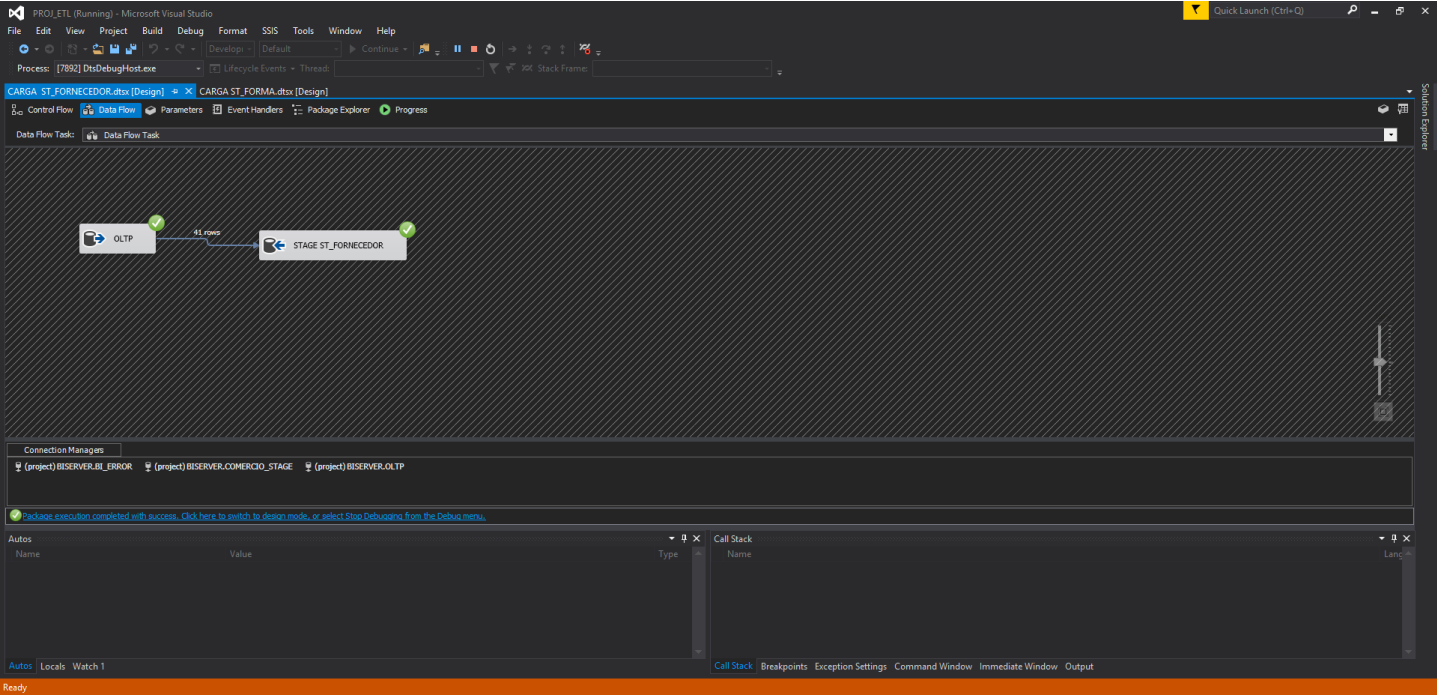


Tabela ST_FORMA

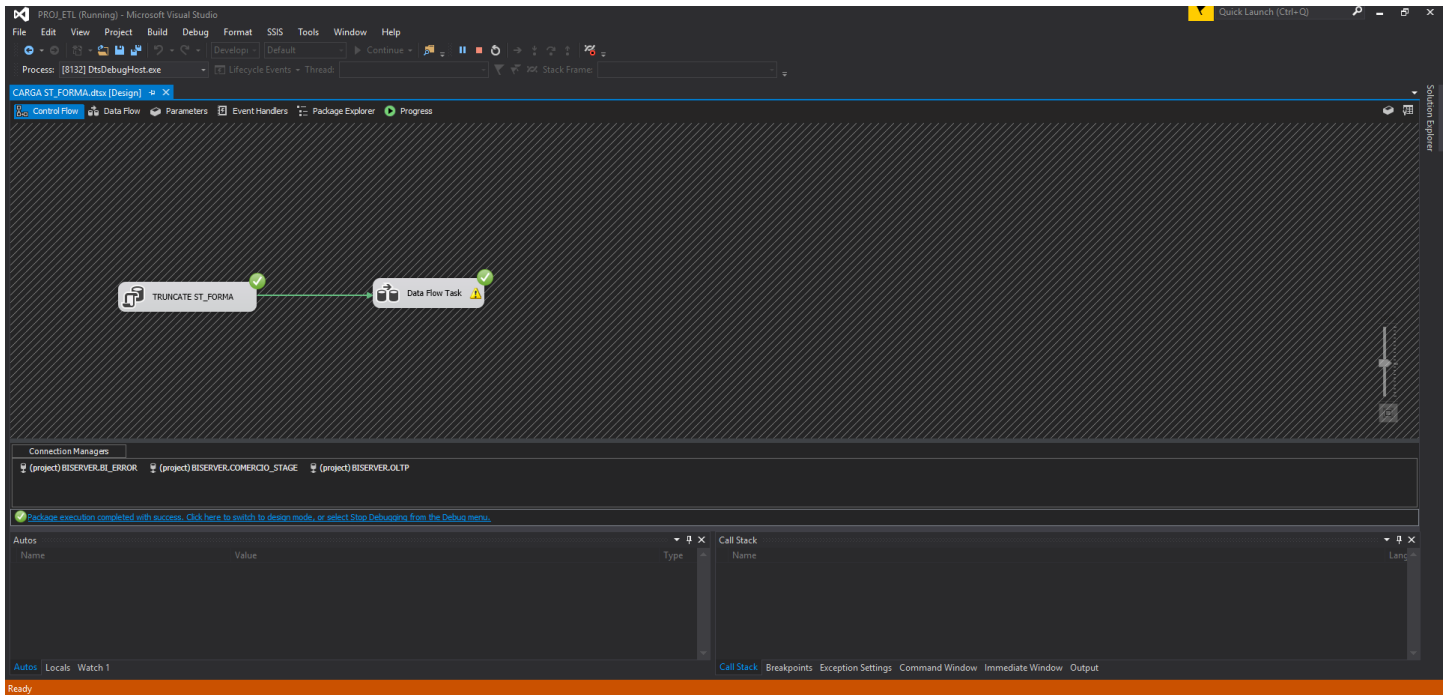


Tabela ST_NOTA

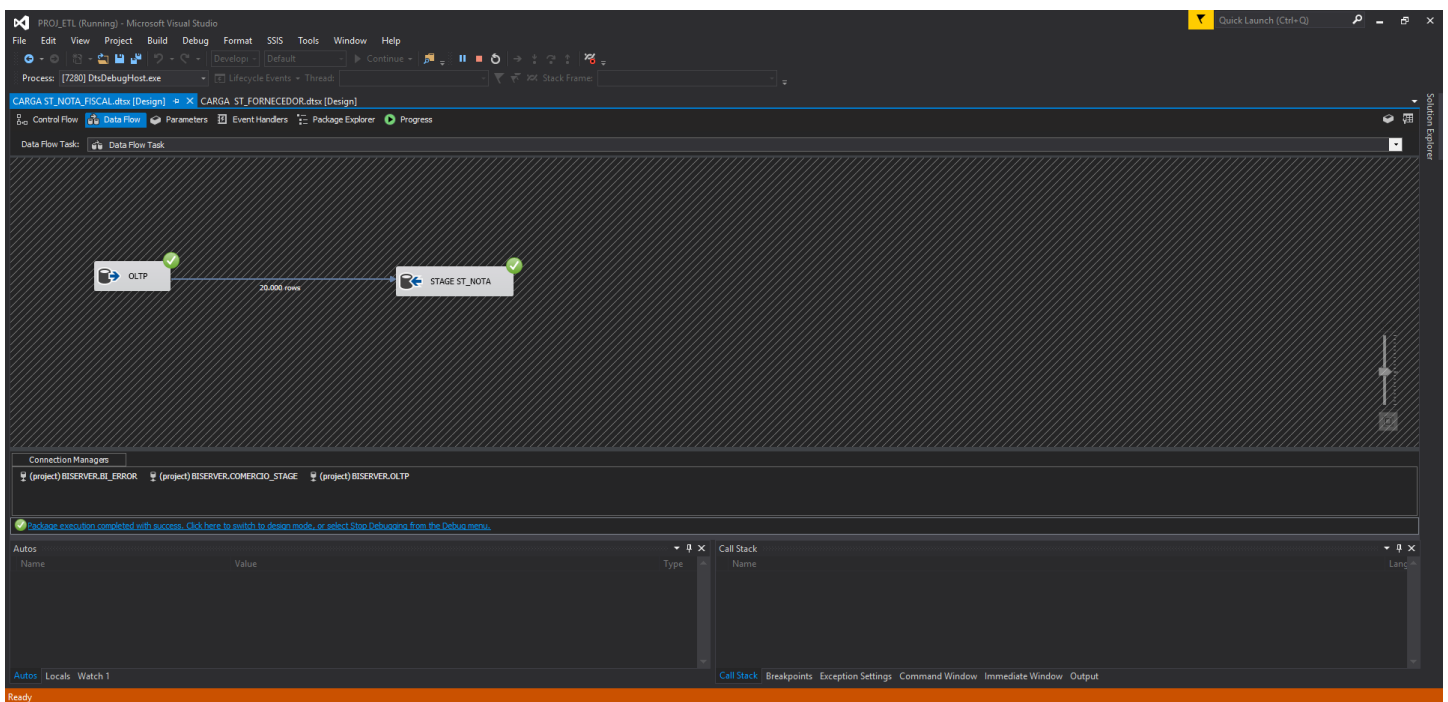


Tabela ST_FATO

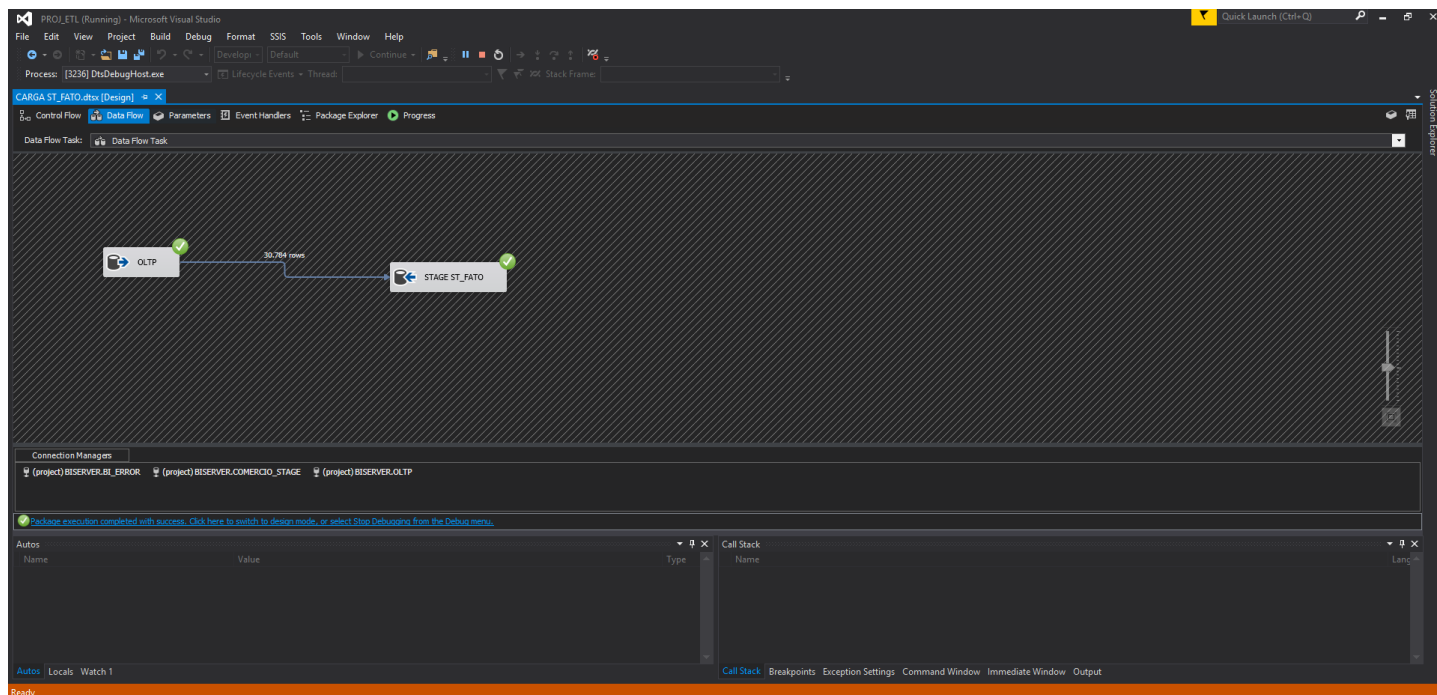


TABELA ST_FATO

A tabela fato contém a agregação de todas as outras tabelas com o objetivo de extrair todas as informações do banco de dados.

Embora não tenha sido criado nenhuma *constraint*, a tabela fato agrega as informações utilizando queries com *JOIN's*, referenciando as Chaves Primárias (*Primary Key* ou *PK*) de cada tabela e de cada linha.

SQLQuery1.sql - BI...CIO_STAGE (sa (54))										
SELECT * FROM ST_FATO										
Results										
IDCLIENTE	IDVENDEDOR	IDPRODUTO	IDFORNECEDOR	IDNOTA	IDFORMA	QUANTIDADE	TOTAL_ITEM	DATA	CUSTO_TOTAL	LUCRO_TOTAL
190	9	62	12	82980	11	1	300.00	2018-07-03	190.00	110.00
776	23	20	4	181700	18	1	40.00	2020-05-20	18.00	22.00
430	6	68	13	136820	25	4	4800.00	2019-02-18	3860.00	940.00
950	11	44	10	55540	4	2	290.00	2018-05-23	152.00	138.00
730	8	105	21	71060	21	3	870.00	2018-01-25	450.00	420.00
232	20	62	12	145040	22	4	1200.00	2019-11-17	760.00	440.00
25	24	150	16	193290	15	1	290.00	2020-12-21	130.00	160.00
722	12	158	40	166480	20	2	218.00	2020-02-23	162.00	56.00
318	1	24	7	199430	6	4	172.00	2020-09-22	92.00	80.00
785	15	1	1	40930	8	3	90.00	2017-02-20	45.00	45.00
81	10	28	7	80120	18	2	290.00	2018-02-10	180.00	110.00
656	20	203	30	137660	4	3	3000.00	2019-02-18	2469.00	531.00
673	23	122	22	88100	13	3	10500.00	2018-04-19	8400.00	2100.00
936	21	107	21	27850	8	2	1520.00	2017-08-11	780.00	740.00
295	22	93	17	58430	9	1	1900.00	2018-02-21	1300.00	600.00
230	23	121	22	156510	4	2	9600.00	2020-12-16	5998.00	3602.00
954	14	96	25	81170	25	4	120.00	2018-11-05	40.00	80.00
51	1	20	4	106040	3	2	80.00	2019-04-25	36.00	44.00
120	19	113	41	138860	19	2	2800.00	2019-01-25	1960.00	840.00
127	21	95	21	95150	25	1	50.00	2018-01-06	25.00	25.00
786	1	55	9	167700	6	3	360.00	2020-07-07	165.00	195.00
26	1	120	18	39590	22	2	5500.00	2017-08-09	4800.00	700.00
243	15	169	13	22030	12	1	1210.00	2017-10-09	763.00	447.00
761	6	89	23	145280	21	2	4600.00	2019-11-14	3600.00	1000.00
684	6	164	14	84390	19	3	6000.00	2018-08-03	4860.00	1140.00
264	21	141	24	7460	10	2	700.00	2017-09-06	406.00	294.00
250	18	184	14	144160	24	2	5600.00	2019-04-03	4372.00	1228.00
859	1	212	32	105420	8	1	360.00	2019-04-25	300.00	60.00
15	20	137	23	94970	19	1	79.00	2018-01-06	60.00	19.00
88	5	138	24	86620	22	4	320.00	2018-12-18	180.00	140.00
72	15	4	1	156820	7	3	72.00	2020-08-11	36.00	36.00

É possível também a criação de uma *View* para extrair a mesma informação. Entretanto, para esse projeto, será utilizada a tabela ST_FATO para facilitar a carga no *Data Warehouse*.

16 - VIEW CARGA ST...CIO_OLTP (sa (53))

SQLQuery2.sql - BI...CIO_STAGE (sa (56))*

14 - CURSOR CAMEL...IO_STAGE (sa (57))

CREATE VIEW RELATORIO_VENDAS AS

SELECT C.IDCLIENTE AS IDCLIENTE,

V.IDVENDEDOR AS IDVENDEDOR,

P.IDPRODUTO AS IDPRODUTO,

FO.IDFORNECEDOR AS IDFORNECEDOR,

N.IDNOTA AS IDNOTA,

IDFORMA AS IDFORMA,

I.QUANTIDADE AS QUANTIDADE,

(I.QUANTIDADE * P.CUSTO_MEDIO) AS CUSTO_TOTAL,

(I.TOTAL - (I.QUANTIDADE * P.CUSTO_MEDIO)) AS LUCRO_TOTAL,

I.TOTAL AS TOTAL_ITEM,

N.DATA AS DATA

FROM NOTA_FISCAL N

INNER JOIN ITEM_NOTA I ON (N.IDNOTA = I.ID_NOTA_FISCAL)

INNER JOIN CLIENTE C ON (C.IDCLIENTE = N.ID_CLIENTE)

INNER JOIN VENDEDOR V ON (V.IDVENDEDOR = N.ID_VENDEDOR)

INNER JOIN PRODUTO P ON (P.IDPRODUTO = ID_PRODUTO)

INNER JOIN FORMA_PAGAMENTO F ON (F.IDFORMA = N.ID_FORMA)

INNER JOIN FORNECEDOR FO ON (FO.IDFORNECEDOR = P.ID_FORNECEDOR)

GO

SELECT * FROM RELATORIO_VENDAS

ORDER BY IDNOTA

GO

100 %

Results

Messages

	IDCLIENTE	IDVENDEDOR	IDPRODUTO	IDFORNECEDOR	IDNOTA	IDFORMA	QUANTIDADE	CUSTO_TOTAL	LUCRO_TOTAL	TOTAL_ITEM	DATA
1	761	21	35	7	1000	25	2	52.00	22.00	74.00	2017-08-25
2	761	21	80	18	1000	25	4	3960.00	3960.00	7920.00	2017-08-25
3	761	21	206	31	1000	25	2	2174.00	826.00	3000.00	2017-08-25
4	816	6	171	13	1010	24	1	3000.00	800.00	3800.00	2017-06-14
5	816	6	66	13	1010	24	3	1155.00	525.00	1680.00	2017-06-14
6	593	21	141	24	1020	2	2	406.00	294.00	700.00	2017-10-22
7	644	6	60	11	1030	24	1	55.00	44.00	99.00	2017-10-23
8	703	21	183	18	1040	5	3	8052.00	1248.00	9300.00	2017-02-18
9	703	21	113	41	1040	5	2	1960.00	840.00	2800.00	2017-02-18
10	632	9	174	14	1050	26	2	200.00	260.00	460.00	2017-08-04
11	632	9	229	35	1050	26	1	1623.00	177.00	1800.00	2017-08-04
12	632	9	35	7	1050	26	3	78.00	33.00	111.00	2017-08-04
13	621	16	145	13	1060	10	3	165.00	69.00	234.00	2017-09-08
14	621	16	130	18	1060	10	4	480.00	480.00	960.00	2017-09-08
15	621	16	9	22	1060	10	4	60.00	60.00	120.00	2017-09-08

Query executed successfully.

CONCLUSÃO DO PROJETO STAGE AREA (COMERCIO_STAGE)

A criação da tabela ST_FATO, conclui então o projeto da *Stage Area*, que é o ponto intermediário entre o ambiente OLTP e o *Data Warehouse*, facilitando as cargas e transformações de dados necessárias sem congestionar os processos do banco de dados transacional e o DW, se tornando o ambiente OLAP necessário para um analista de BI trabalhar com mais segurança e independência.

Os pacotes de ETL criados no SSIS, serão automatizados consoante a entrada de novos dados do sistema OLTP. Este processo poderá ser agendado (a noite, por exemplo) para que seja feita a ligação entre os sistemas e os dados estejam disponíveis para operação e consulta.

PACOTES DE ETL DO INTEGRATION SERVICES (SSIS)

O processo então é propagado para as demais tabelas da base de dados, criando um projeto de pacotes de *ETL* específico para cada uma das tabelas.

Então, os pacotes se resumem em um projeto que poderá ser automatizado mediante nova entrada de dados no banco transacional, *OLTP*.

