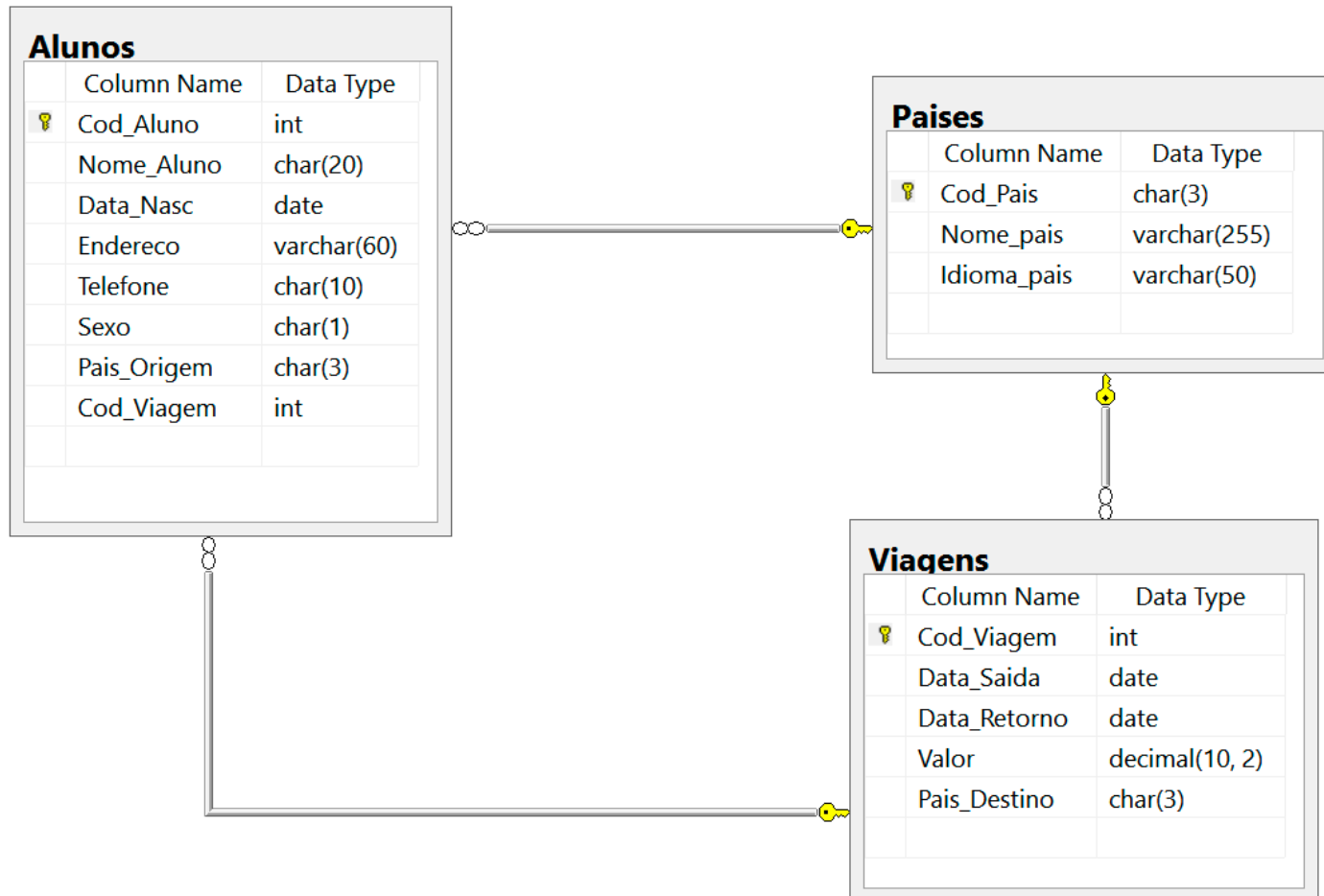




- **Aula 014 – Linguagem SQL**
 - Funções para manipulação de dados.
 - Funções para manipulação de data, hora e strings.
 - Funções para conversão de dados.



■ Banco de dados INTERCAMBIO



■ Operadores de junção de SQL

Tipo de Junção	Descrição
CROSS JOIN	Retorna o produto cartesiano de duas tabelas.
JOIN ON	Retorna apenas as linhas que atendam à condição de junção descrita na cláusula ON.
NATURAL JOIN	Retorna apenas as linhas com valores correspondentes nas cláusulas correspondentes
JOIN USING	Retorna apenas as linhas com valores correspondentes nas colunas indicadas pela cláusula USING.
LEFT OUTER JOIN	Retorna linhas com valores correspondentes e inclui todas as colunas da tabela à esquerda de LEFT.
RIGHT OUTER JOIN	Retorna linhas com valores correspondentes e inclui todas as colunas da tabela à direita de RIGHT.
FULL OUTER JOIN	Retorna linhas com valores correspondentes e inclui todas as colunas de ambas as tabelas.



- **Funções para manipulação de dados**
 - Os dados de um banco de dados são a base de informações fundamentais de negócios e a geração de informações a partir desses dados costuma exigir muitas manipulações.
 - Para essa tarefa, a linguagem SQL possui um conjunto de funções que permitem a manipulação de caracteres, a manipulação de data e hora, a realização de operações matemáticas, além da conversão de valores.



■ Funções para operações matemáticas

Função	Descrição
ABS	Retorna o valor absoluto da expressão numérica especificada.
CEILING	Retorna o menor inteiro maior que ou igual a expressão numérica especificada
FLOOR	Retorna o maior inteiro menor ou igual à expressão numérica especificada.
EXP	Retorna o valor exponencial do número de Euler.
POWER	Retorna o valor da expressão especificada elevada à potência especificada.
RAND	Retorna um valor pseudoaleatório entre 0 e 1.
ROUND	Retorna um valor numérico, arredondado, para o comprimento ou precisão especificados.
PI	Retorna o valor do número PI.



■ Funções para operações matemáticas

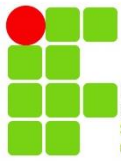
Função	Descrição
LOG	Retorna o logaritmo natural da expressão especificada.
LOG10	Retorna o logaritmo de base da expressão especificada.
SQRT	Retorna a raiz quadrada da expressão numérica especificada.
SIGN	Retorna o sinal positivo (+1), zero (0) ou sinal negativo (-1) da expressão especificada.
SQUARE	Retorna o quadrado da expressão numérica especificada.
DEGREES	Retorna o ângulo correspondente em graus para um ângulo especificado em radiano.
RADIANS	Retorna o ângulo correspondente em radianos para um ângulo especificado em graus.
SIN	Retorna o seno trigonométrico do ângulo especificado em radianos.



■ Funções para operações matemáticas

Função	Descrição
<code>COS</code>	Retorna o cosseno trigonométrico do ângulo especificado em radianos.
<code>ASIN</code>	Retorna o arco seno do ângulo especificado em radianos.
<code>ACOS</code>	Retorna o arco cosseno do ângulo especificado em radianos.
<code>TAN</code>	Retorna a tangente do ângulo especificado em radianos.
<code>COT</code>	Retorna a cotangente do ângulo especificado em radianos.
<code>ATAN</code>	Retorna o arco tangente do ângulo especificado em radianos.
<code>ATN2</code>	Retorna o ângulo, em radianos, entre o eixo x positivo e o raio a partir da origem até o ponto (y, x), em que x e y são os valores das duas expressões flutuantes especificadas.





■ Funções para operações matemáticas

SELECT '3.1415'	AS 'PI',
PI()	AS 'PI',
ABS(-3.1415)	AS 'ABS',
CEILING(3.1415)	AS 'CEILING',
FLOOR(3.1415)	AS 'FLOOR',
EXP(1.0)	AS 'EXP',
POWER(2, 3.0)	AS 'POWER',
RAND(5)	AS 'RAND',
ROUND(PI(), 2)	AS 'ROUND',
SQRT(100)	AS 'SQRT',
SIGN(-1)	AS 'SIGN',
SQUARE(3)	AS 'SQUARE'



■ Funções para operações matemáticas

-- Exibe informações sobre o valor das viagens dos alunos.

```
SELECT V.Cod_Viagem    AS 'Código da Viagem',  
       A.Nome_Aluno    AS 'Nome do Aluno',  
       V.Valor          AS 'Preço da Viagem',  
       V.Valor * 0.05   AS 'Desconto de 5%',  
       V.Valor * 0.95   AS 'Total a Pagar',  
       ➡ ROUND(V.Valor * 0.95, 1) AS 'Total a Pagar  
(ROUND) '  
FROM Viagens V INNER JOIN ALUNOS A  
     ON V.Cod_Viagem = A.Cod_Viagem  
GO
```



■ Funções para operações matemáticas

Código da Viagem	Nome do Aluno	Preço da Viagem	Desconto de 5%	Total a Pagar	Total a Pagar (ROUND)

1	Maria Cristina	5350.00	267.5000	5082.5000	5082.5000
2	Ana Paula Lima	8900.00	445.0000	8455.0000	8455.0000
3	Carlos Renato	18525.35	926.2675	17599.0825	17599.1000
4	Hugo Silva	12350.25	617.5125	11732.7375	11732.7000
5	Marcos Antônio	7520.00	376.0000	7144.0000	7144.0000
6	Gislaine Silva	14255.35	712.7675	13542.5825	13542.6000
7	Antônio Pereira	3250.00	162.5000	3087.5000	3087.5000
8	Jair Lopes	9500.00	475.0000	9025.0000	9025.0000
9	Miguel Firmino	9250.00	462.5000	8787.5000	8787.5000
.....					
2	Heidi Lima	8900.00	445.0000	8455.0000	8455.0000
8	Amílcar Júnior	9500.00	475.0000	9025.0000	9025.0000
11	Alexandro Duarte	12355.00	617.7500	11737.2500	11737.3000
16	Maurício dos Santos	22850.25	1142.5125	21707.7375	21707.7000
17	Mary Ann Duarte	18600.10	930.0050	17670.0950	17670.1000
18	Gabriela Pereira	19000.50	950.0250	18050.4750	18050.5000
14	André César	15300.00	765.0000	14535.0000	14535.0000
14	Edson Lopes	15300.00	765.0000	14535.0000	14535.0000

(50 row(s) affected)



- Funções para manipulação de data e hora – precisão superior

Função	Descrição
<code>SYSDATETIME</code>	Retorna um valor do tipo <code>DATETIME2</code> que contém a data e hora do computador no qual a instância do SQL Server está em execução. O deslocamento de fuso horário não está incluído.
<code>SYSDATETIMEOFFSET</code>	Retorna um valor do tipo <code>DATETIMEOFFSET</code> que contém a data e hora do computador no qual a instância do SQL Server está em execução. O deslocamento de fuso horário não está incluído.
<code>SYSUTCDATETIME</code>	Retorna um valor do tipo <code>DATETIME2</code> que contém a data e hora do computador no qual a instância do SQL Server está em execução. A data e hora é retornada como hora UTC (Coordinated Universal Time).



- Funções para manipulação de data e hora – precisão superior

Função	Descrição
<code>CURRENT_TIMESTAMP</code>	Retorna um valor do tipo <code>DATETIME</code> que contém a data e hora do computador no qual a instância do SQL Server está em execução. O deslocamento de fuso horário não está incluído.
<code>GETDATE</code>	Retorna um valor do tipo <code>DATETIME</code> que contém a data e hora do computador no qual a instância do SQL Server está em execução. O deslocamento de fuso horário não está incluído.
<code>GETUTCDATE</code>	Retorna um valor do tipo <code>DATETIME</code> que contém a data e hora do computador no qual a instância do SQL Server está em execução. A data e hora é retornada como hora UTC (Coordinated Universal Time).



- Funções para manipulação de data e hora – precisão superior

-- Exemplo de funções com precisão superior e inferior

```
SELECT SYSDATETIME()           AS 'SYSDATETIME',  
       SYSDATETIMEOFFSET()    AS 'SYSDATETIMEOFFSET',  
       SYSUTCDATETIME()       AS 'SYSUTCDATETIME',  
       CURRENT_TIMESTAMP      AS 'CURRENT_TIMESTAMP',  
       → GETDATE()            AS 'GETDATE',  
       GETUTCDATE()           AS 'GETUTCDATE'
```

GO



■ Funções que obtêm uma parte de uma data

Função	Descrição
<code>DATENAME (parte, data)</code>	Retorna uma cadeia de caracteres que representa a parte da data especificada.
<code>DATEPART (parte, data)</code>	Retorna um inteiro que representa a parte da data especificada.
<code>DAY (data)</code>	Retorna um inteiro que representa a parte do dia da data especificada.
<code>MONTH (data)</code>	Retorna um inteiro que representa a parte do mês da data especificada.
<code>YEAR (data)</code>	Retorna um inteiro que representa a parte do ano da data especificada.



■ Funções que obtêm uma parte de uma data

Parte	Abreviação	Parte	Abreviação
year	y, yyyy	millisecond	ms
quarter	q, qq	microsecond	mcs
month	m, mm	nanosecond	ns
dayofyear	dy	TZoffset	tz
day	d, dd	ISO_WEEK	isowk, isoww
week	wk, ww		
weekday	dw		
hour	hh		
minute	n, min		
second	s, ss		



■ Funções que obtêm uma parte de uma data

-- Exemplo de funções que obtêm a parte de uma data

```
SELECT  Cod_Aluno AS 'Código',  
        Data_Nasc AS 'Data de Nascimento',  
        DAY(Data_Nasc) AS 'Dia do Nascimento',  
        MONTH(Data_Nasc) AS 'Mês do Nascimento',  
        YEAR(Data_Nasc) AS 'Ano do Nascimento',  
        DATEPART(WEEK, Data_Nasc) AS 'Semana do  
Nascimento',  
        DATEPART(WEEKDAY, Data_Nasc) AS 'Dia da  
Semana do Nascimento'  
FROM ALUNOS
```



■ Funções que obtêm uma parte de uma data

Código	Data de Nascimento	Dia do Nascimento	Mês do Nascimento	Ano do Nascimento	Semana do Nascimento	Dia da Semana do Nascimento
1	1985-10-12	12	10	1985	41	7
2	1980-12-01	1	12	1980	49	2
3	1979-05-10	10	5	1979	19	5
4	1975-10-02	2	10	1975	40	5
5	1985-10-23	23	10	1985	43	4
6	1983-11-06	6	11	1983	46	1
7	1979-06-18	18	6	1979	25	2
8	1982-12-14	14	12	1982	51	3
9	1982-05-12	12	5	1982	20	4
38	1983-05-15	15	5	1983	21	1
39	1982-06-15	15	6	1982	25	3
40	1978-04-17	17	4	1978	16	2
41	1984-04-15	15	4	1984	16	1
42	1978-10-11	11	10	1978	41	4
43	1978-08-11	11	8	1978	32	6
44	1976-04-10	10	4	1976	15	7
45	1974-05-29	29	5	1974	22	4
46	1988-04-13	13	4	1988	16	4
47	1989-04-13	13	4	1989	15	5
48	1983-07-29	29	7	1983	31	6
49	1984-12-30	30	12	1984	53	1
50	1979-01-11	11	1	1979	2	5

(50 row(s) affected)



- **Funções que obtêm valores de data e hora de suas partes**

Função	Descrição
<code>DATEFROMPARTS (ano, mês, dia)</code>	Retorna um valor do tipo DATE valor para o ano, mês e dia especificados.
<code>DATETIME2FROMPARTS (ano, mês, dia, hora, minuto, segundo, frações, precisão)</code>	Retorna um valor do tipo DATETIME2 para a data e hora especificadas e com a precisão especificada.
<code>DATETIMEFROMPARTS (ano, mês, dia, hora, minuto, segundo, milissegundo)</code>	Retorna um valor do tipo DATETIME para a data e hora especificadas e com a precisão especificada.



- **Funções que obtêm valores de data e hora de suas partes**

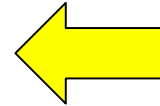
Função	Descrição
<code>DATETIMEOFFSETFROMPARTS (ano, mês, dia, hora, minuto, segundos, frações, offset_hora, offset_minuto, precisão)</code>	Retorna um valor do tipo DATETIMEOFFSET para a data e hora especificada e com os deslocamentos e precisão especificados.
<code>SMALLDATETIMEFROMPARTS (ano, mês, dia, hora, minuto)</code>	Retorna um valor do tipo SMALLDATETIME para a data e hora especificada.
<code>TIMEFROMPARTS (hora, minuto, segundos, frações, precisão)</code>	Retorna um valor do tipo TIME para a hora especificada e com a precisão especificada.



- Funções que obtêm valores de data e hora de suas partes

-- Declaração de variáveis em T-SQL

```
DECLARE @dia    AS INT,  
        @mes    AS CHAR(20),  
        @ano    AS INT,  
        @data1  AS DATE,  
        @data2  AS DATETIME
```



-- Atribuindo valores

```
SET @dia = DAY(GETDATE())  
SET @mes = MONTH(GETDATE())  
SET @ano = YEAR(GETDATE())
```



- Funções que obtêm valores de data e hora de suas partes

-- Cria uma nova data

```
SET @data1 = DATEFROMPARTS(@ano, @mes, @dia)
```

```
SET @data2 = DATETIMEFROMPARTS(@ano, @mes,  
@dia, 0, 0, 0, 0)
```

```
SELECT @dia      AS 'Dia',  
       @mes      AS 'Mês',  
       @ano      AS 'Ano',  
       @data1    AS 'Data 1',  
       @data2    AS 'Data 2'
```



- **Funções que obtêm valores de data e hora de suas partes**
 - Todo o trecho que utiliza as variáveis que foram definidas em uma instrução `DECLARE` deve ser selecionado durante a execução de um código, caso contrário ocorrerá um erro:

Msg 137, Level 15, State 2, Line 335

Must declare the scalar variable "@dia".

- **Resultado correto:**

Dia	Mês	Ano	Data 1	Data 2
26	8	2018	2018-08-26	2018-08-26 00:00:00.000



- **Funções que obtêm a diferença entre datas e horas**

Função	Descrição
<code>DATEDIFF (parte, data_inicial, data_final)</code>	Retorna o valor da data ou hora que representa a diferença da parte entre duas datas especificadas.
<code>DATEDIFF_BIG (parte, data_inicial, data_final)</code>	Retorna o valor da data ou hora que representa a diferença da parte entre duas datas especificadas.



- **Funções que obtêm a diferença entre datas e horas**

- Declara duas datas

```
DECLARE @data1 AS DATE,  
        @data2 AS DATE
```

- Altera o formato de entrada das datas

```
SET DATEFORMAT DMY ←
```

- Atribui alguns valores

```
SET @data1 = '02/02/2017'
```

```
SET @data2 = GETDATE()
```



- **Funções que obtêm a diferença entre datas e horas**

-- Utiliza DATEDIFF para calcular a diferença entre as datas

```
SELECT @data1 AS 'Data Inicial',  
       @data2 AS 'Data de Hoje',  
       DATEDIFF(DAY, @data1, @data2) AS  
'Qtd. Dias',  
       DATEDIFF(MONTH, @data1, @data2) AS  
'Qtd. Meses',  
       DATEDIFF(HOUR, @data1, @data2) AS  
'Qtd. Horas'
```

GO



- Funções que obtêm a diferença entre datas e horas

Data Inicial	Data de Hoje	Qtd. Dias	Qtd. Meses	Qtd. Horas
-----	-----	-----	-----	-----
2017-02-02	2018-08-26	570	18	13680

(1 row(s) affected)



- **Funções que modificam uma data e os valores da hora**

Função	Descrição
<code>DATEADD (parte, número, data)</code>	Retorna um valor do tipo DATETIME adicionando um número na parte da data especificada.
<code>EOMONTH (data_inicial, mês_de_adição]</code>	Retorna o último dia do mês que contém a data especificada com um deslocamento opcional.
<code>SWITCHOFFSET (DATETIME OFFSET, fuso_horário)</code>	Altera o fuso-horário de uma data especificada e preserva o valor UTC.
<code>TODATETIMEOFFSET (expressão, fuso_horário)</code>	Transforma um valor do tipo DATETIME2 em um valor do tipo DATETIMEOFFSET .



- **Funções que modificam uma data e os valores da hora**

-- Modificação de data e valores da hora

```
SELECT GETDATE() AS 'Data atual',  
       DATEADD(MONTH, 5, GETDATE()) AS  
       'Próximos 5 meses',  
       EOMONTH(GETDATE(), 5) AS 'Final do  
mês (daqui 5 meses)',  
       SWITCHOFFSET(GETDATE(), '+10:00') AS  
       'Alteração de fuso-horário (+10 horas)'
```

GO



- Funções que modificam uma data e os valores da hora

Data atual

Próximos 5 meses

2018-08-26 18:24:13.413 2019-01-26 18:24:36.413

Final do mês (daqui 5 meses) Alteração de fuso-horário

2019-01-31

2018-08-27 04:24:13.413 +10:00

(1 row(s) affected)



- **Funções que definem ou obtêm um formato de sessão**

Função	Descrição
<code>@@DATEFIRST</code>	Retorna o valor atual, para uma sessão de <code>SET DATEFIRST</code> .
<code>SET DATEFIRST {número}</code>	Define o primeiro dia da semana como um número de 1 a 7.
<code>SET DATEFORMAT {formato}</code>	Define a ordem das partes de uma data (dia/mês/ano).
<code>@@LANGUAGE</code>	Retorna o nome do idioma que está sendo usado.
<code>SET LANGUAGE { [N] 'language' }</code>	Define o ambiente de idioma para as mensagens do sistema e da sessão.
<code>sp_helplanguage [[@language =] 'language']</code>	Retorna informações sobre formatos de data de todos os idiomas com suporte.



■ Formatos para o primeiro dia da semana

Valor	Primeiro dia da semana
1	Segunda-feira
2	Terça-feira
3	Quarta-feira
4	Quinta-feira
5	Sexta-feira
6	Sábado
7	Domingo (Padrão, Inglês dos EUA)



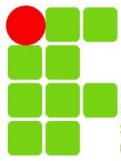
■ Formatos para o primeiro dia da semana

-- Exibe a configuração atual idioma e do
-- primeiro dia da semana

```
SELECT  @@LANGUAGE  AS 'Idioma Utilizado',  
        @@DATEFIRST AS 'Primeiro dia da  
semana'  
GO
```

Idioma Utilizado	Primeiro dia da semana
-----	-----
us_english	7





■ Formatos para o primeiro dia da semana

-- Demonstra a utilização de SELECT... CASE

```
SELECT @@LANGUAGE AS 'Idioma Utilizado',  
CASE  
    WHEN @@DATEFIRST = 1 THEN 'Segunda-feira'  
    WHEN @@DATEFIRST = 2 THEN 'Terça-feira'  
    WHEN @@DATEFIRST = 3 THEN 'Quarta-feira'  
    WHEN @@DATEFIRST = 4 THEN 'Quinta-feira'  
    WHEN @@DATEFIRST = 5 THEN 'Sexta-feira'  
    WHEN @@DATEFIRST = 6 THEN 'Sábado-feira'  
    WHEN @@DATEFIRST = 7 THEN 'Domingo'  
END AS 'Primeiro dia da semana'
```



- **Informações sobre o idioma do servidor**
 - Retorna informações sobre todos
 - os idiomas disponíveis no servidor

```
SELECT * FROM SYS.syslanguages  
  
GO
```



■ Informações sobre o idioma do servidor

-- Retorna informações sobre alguns idiomas

```
SELECT  langid          AS 'ID do idioma',  
        dateformat      AS 'Formato de data',  
        datefirst       AS 'Primeiro dia da semana',  
        name            AS 'Nome do idioma',  
        alias           AS 'Nome alternativo do idioma',  
        months          AS 'Nomes dos meses',  
        shortmonths     AS 'Abreviatura dos meses',  
        days            AS 'Nomes dos dias'  
  
FROM SYS.syslanguages  
  
WHERE alias IN ('English', 'Brazilian', 'German', 'Japanese',  
               'Russian')  
  
GO
```



■ Informações sobre o idioma do servidor

ID do idioma	Formato de data	Primeiro dia da semana	Nome do idioma	Nome alternativo do idioma
0	mdy	7	us_english	English
1	dmy	1	Deutsch	German
3	ymd	7	日本語	Japanese
21	dmy	1	русский	Russian
27	dmy	7	Português (Brasil)	Brazilian

Nomes dos meses

January, February, March, April, May, June, July, August, September, October, November, December
 Januar, Februar, März, April, Mai, Juni, Juli, August, September, Oktober, November, Dezember
 01, 02, 03, 04, 05, 06, 07, 08, 09, 10, 11, 12
 Январь, Февраль, Март, Апрель, Май, Июнь, Июль, Август, Сентябрь, Октябрь, Ноябрь, Декабрь
 Janeiro, Fevereiro, Março, Abril, Maio, Junho, Julho, Agosto, Setembro, Outubro, Novembro, Dezembro

Abreviatura dos meses

Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, Dec
 Jan, Feb, Mär, Apr, Mai, Jun, Jul, Aug, Sep, Okt, Nov, Dez
 01, 02, 03, 04, 05, 06, 07, 08, 09, 10, 11, 12
 янв, фев, мар, апр, май, июн, июл, авг, сен, окт, ноя, дек
 Jan, Fev, Mar, Abr, Mai, Jun, Jul, Ago, Set, Out, Nov, Dez

Nomes dos dias

Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday
 Montag, Dienstag, Mittwoch, Donnerstag, Freitag, Samstag, Sonntag
 月曜日, 火曜日, 水曜日, 木曜日, 金曜日, 土曜日, 日曜日
 понедельник, вторник, среда, четверг, пятница, суббота, воскресенье
 Segunda-Feira, Terça-Feira, Quarta-Feira, Quinta-Feira, Sexta-Feira, Sábado, Domingo



- **Informações sobre o idioma do servidor**

-- Retorna informações sobre alguns idiomas

```
EXEC SP_HELPLANGUAGE [Brazilian]
```

```
GO
```

```
EXEC SP_HELPLANGUAGE [Japanese]
```

```
GO
```

```
EXEC SP_HELPLANGUAGE [English]
```

```
GO
```



■ Estilos de data e hora (SQL Server)

Formato	Padrão	Exemplo
0 ou 100	Padrão para DATETIME e SMALLDATETIME	mês dd aaaa hh:miAM (ou PM)
1 ou 101	EUA	1 = mm/dd/aa 101 = mm/dd/aaaa
2 ou 102	ANSI	2 = aa.mm.dd 102 = aaaa.mm.dd
3 ou 103	Inglês / Francês	3 = dd/mm/aa 103 = dd/mm/aaaa
4 ou 104	Alemão	4 = dd.mm.aa 104 = dd.mm.aaaa
5 ou 105	Italiano	5 = dd-mm-aa 105 = dd-mm-aaaa

■ Estilos de data e hora (SQL Server)

Formato	Padrão	Exemplo
6 ou 106	---	6 = dia mês aa 106 = dd mês aaaa
7 ou 107	---	7 = Mês dd, aa 107 = Mês dd, aaaa
8 ou 108	---	hh:mi:ss
9 ou 109	Padrão + milissegundos	mês dd aaaa hh:mi:ss:mmm AM (ou PM)
10 ou 110	EUA	10 = mm-dd-aa 110 = mm-dd-aaaa
11 ou 111	Japão	11 = aa/mm/dd 111 = aaaa/mm/dd

■ Estilos de data e hora (SQL Server)

Formato	Padrão	Exemplo
12 ou 112	ISO	12 = aammdd 112 = aaaammdd
13 ou 113	Padrão Europeu + milissegundos	dd mês aaaa hh:mi:ss:mmm (24h)
14 ou 114	---	hh:mi:ss:mmm (24h)



■ Estilos de data e hora (SQL Server)

-- Declara uma variável para armazenar a data atual

```
DECLARE @data DATETIME
```

-- Atribui o valor da data atual

```
SET @data = GETDATE()
```

-- Exibe informações sobre a data atual

```
SELECT @data AS 'Data Atual',  
       DATEPART(DAY, @data) AS 'Dia do Mês',  
       ➡ DATENAME(DW, @data) AS 'Dia da Semana',  
       DATEPART(MONTH, @data) AS 'Mês',  
       ➡ DATENAME(MONTH, @data) AS 'Nome do Mês',  
       DATEPART(YEAR, @data) AS 'Ano'
```



■ Estilos de data e hora (SQL Server)

Data Atual		Dia do Mês	Dia da Semana	Mês	Nome do Mês	Ano
2018-08-26 18:51:25.217	26		Sunday	8	August	2018

(1 row(s) affected)



Os nomes dos dias da semana e dos meses dependem do idioma utilizado



■ Estilos de data e hora (SQL Server)

-- Declara uma variável para armazenar a data atual

```
DECLARE @data DATETIME
```

-- Atribui o valor da data atual

```
SET @data = GETDATE()
```

-- Outra maneira de exibir informações sobre a data atual

```
SELECT @data AS 'Data Atual',  
       DATENAME(DW, @data) AS 'Dia da Semana',  
       DATENAME(WK, @data) AS 'Semana do Ano',  
       DATENAME(M, @data) AS 'Nome do Mês',  
       DATENAME(D, @data) AS 'Dia do Mês',  
       DATENAME(DY, @data) AS 'Dia do Ano'
```



■ Estilos de data e hora (SQL Server)

Data Atual	Dia da Semana	Semana do Ano	Nome do Mês	Dia do Mês	Dia do Ano
-----	-----	-----	-----	-----	-----
2018-08-26 18:56:31.937	Sunday	35	August	26	238

(1 row(s) affected)



■ Estilos de data e hora (SQL Server)

-- Altera o idioma para o Português do Brasil

```
SET LANGUAGE Brazilian
```

```
GO
```

-- Declara uma variável para armazenar a data atual

```
DECLARE @data DATETIME
```

-- Atribui o valor da data atual

```
SET @data = GETDATE()
```

-- Exibe o idioma, dia da semana e nome do mês

```
SELECT  @@LANGUAGE          AS 'Idioma',  
        DATENAME(DW, @data) AS 'Dia da Semana',  
        DATENAME(M, @data)  AS 'Nome do Mês'
```



```
GO
```



■ Estilos de data e hora (SQL Server)

Idioma	Dia da Semana	Nome do Mês
Português (Brasil)	Domingo	Agosto

(1 row(s) affected)



- Funções para validar uma data e valores de tempo

Função	Descrição
ISDATE (expressão)	Determina se um valor do tipo DATETIME ou SMALLDATETIME fornecido como parâmetro é uma data válida ou um valor de tempo.



■ Estilos de data e hora (SQL Server)

-- Altera o formato de data e hora -> DMY

SET DATEFORMAT DMY 

GO

-- Utiliza ISDATE e um IF-ELSE para validar uma data

IF ISDATE('20/01/2015 00:10:50.000') = 1

PRINT 'Data válida!' 

ELSE

PRINT 'Data inválida!'

GO



■ Estilos de data e hora (SQL Server)

-- Altera o formato de data e hora -> MDY

SET DATEFORMAT MDY 

GO

-- Utiliza ISDATE e um IF-ELSE para validar uma data

IF ISDATE('20/01/2015 00:10:50.000') = 1

PRINT 'Data válida!'

ELSE

PRINT 'Data inválida!' 

GO



■ Funções para manipulação de strings

Função	Descrição
<code>ASCII (expressão)</code>	Retorna o valor do código ASCII do caractere mais à esquerda de uma expressão de caracteres.
<code>CHAR(inteiro)</code>	Converte um valor numérico inteiro para o seu valor ASCII correspondente.
<code>CHARINDEX (expressão, texto [, posição_inicial])</code>	Procura uma expressão dentro de um texto e retorna sua posição inicial.
<code>NCHAR(inteiro)</code>	Retorna o código Unicode para o inteiro fornecido como parâmetro.
<code>UNICODE (expressão)</code>	Retorna o valor do código UNICODE do caractere mais à esquerda de uma expressão de caracteres.

■ Funções para manipulação de strings

-- Demonstra a utilização de algumas funções que manipulam o
-- código ASCII e o código Unicode

```
SELECT ASCII('IFSP')      AS 'ASCII - IFSP',  
       UNICODE('IFSP')   AS 'UNICODE - IFSP',  
       CHAR(73)          AS 'CHAR - 73',  
       NCHAR(73)         AS 'NCHAR - 73',  
       ASCII(N'Бразилия') AS 'ASCII - Бразилия',  
       UNICODE(N'Бразилия') AS 'UNICODE - Бразилия',  
       CHAR(1041)        AS 'CHAR - 1041',  
       NCHAR(1041)       AS 'NCHAR - 1041',  
       CHARINDEX('IFSP', 'Câmpus do IFSP') AS 'CHARINDEX'
```



■ Funções para manipulação de strings

ASCII - IFSP UNICODE - IFSP CHAR - 73 NCHAR - 73

73 73 I I

ASCII - Бразилия UNICODE - Бразилия CHAR - 1041 NCHAR - 1041

63 1041 NULL Б

CHARINDEX

11



■ Funções para manipulação de strings

Função	Descrição
<code>SPACE (inteiro)</code>	Retorna uma cadeia de caracteres de espaços repetidos.
<code>QUOTENAME (expressão [, delimitadores])</code>	Retorna uma cadeia de caracteres Unicode com os delimitadores adicionados para tornar a cadeia de caracteres de entrada um identificador delimitado válido do SQL Server.
<code>STR(valor_real [, tamanho [, decimais]])</code>	Converte dados numéricos para o formato de uma cadeia de caracteres.
<code>LEN (expressão)</code>	Retorna o número de caracteres da expressão da cadeia de caracteres especificada, excluindo espaços em branco à direita.

■ Funções para manipulação de strings

-- Demonstra a utilização do SPACE, QUOTENAME, STR e LEN

```
SELECT 'Paulo' + 'Giovani'          AS 'SPACE 1',  
       'Paulo' + ' ' + 'Giovani' AS 'SPACE 2',  
       'Paulo' + SPACE(10) + 'Giovani' AS 'SPACE 3',  
       QUOTENAME('Paulo Giovanni', '{') AS 'QUOTENAME 1',  
       QUOTENAME('Paulo Giovanni', '"') AS 'QUOTENAME 2',  
       QUOTENAME('Paulo Giovanni', '[') AS 'QUOTENAME 3',  
       → { STR(100)          AS 'STR 1',  
            STR(100.0)       AS 'STR 2',  
            STR(100.45, 6, 2) AS 'STR 3',  
            LEN('Paulo Giovanni') AS 'LEN 1'
```

GO



■ Funções para manipulação de strings

SPACE 1	SPACE 2	SPACE 3
-----	-----	-----
PauloGiovani	Paulo Giovanni	Paulo Giovanni

QUOTENAME 1	QUOTENAME 2	QUOTENAME 3
-----	-----	-----
{Paulo Giovanni}	"Paulo Giovanni"	[Paulo Giovanni]

STR 1	STR 2	STR 3	LEN 1
-----	-----	-----	-----
100	100	100.45	13



■ Funções para manipulação de strings

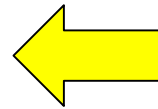
Função	Descrição
<code>PATINDEX (padrão, expressão)</code>	Retorna a posição inicial da primeira ocorrência de um padrão em uma expressão específica ou zeros, se o padrão não for encontrado, em todos os tipos de dados de caractere e de texto válidos.
<code>SOUNDEX (expressão)</code>	Retorna um código de quatro caracteres (SOUNDEX) para avaliar a semelhança de duas cadeias de caracteres.
<code>DIFFERENCE (expressão_1, expressão_2)</code>	Retorna um valor de inteiro que indica a diferença entre os valores SOUNDEX de duas expressões de caractere.
<code>FORMAT (valor, formato [, cultura])</code>	Retorna um valor formatado com o formato e a cultura (opcional) especificados no SQL Server (a partir da versão 2012).

■ Funções para manipulação de strings

-- Demonstra a utilização de PATINDEX

```
SELECT PATINDEX('mp', 'Campus Campos do  
Jordão') AS 'PATINDEX 1',  
        PATINDEX('%mp%', 'Campus Campos do  
Jordão') AS 'PATINDEX 2'  
GO
```

PATINDEX 1	PATINDEX 2
-----	-----
0	3



■ Funções para manipulação de strings

-- Demonstra a utilização de SOUNDEX e DIFFERENCE

```
SELECT SOUNDEX('Paulo') AS 'SOUNDEX - Paulo',  
       SOUNDEX('Paul') AS 'SOUNDEX - Paul',  
       SOUNDEX('Cris') AS 'SOUNDEX - Cris',  
       DIFFERENCE('Paulo', 'Paul') AS 'DIFF 1',  
       DIFFERENCE('Paulo', 'Cris') AS 'DIFF 2'
```

GO

SOUNDEX - Paulo	SOUNDEX - Paul	SOUNDEX - Cris	DIFFERENCE 1	DIFFERENCE 2
-----	-----	-----	-----	-----
P400	P400	C620	4	2



■ Funções para manipulação de strings

-- Exibe a data atual, formatada em vários idiomas

```
DECLARE @d DATETIME = GETDATE()
```

```
SELECT FORMAT(@d, 'D', 'en-US') AS 'Inglês Americano',  
        FORMAT(@d, 'D', 'en-gb') AS 'Inglês Britânico',  
        FORMAT(@d, 'D', 'de-de') AS 'Alemão',  
        FORMAT(@d, 'D', 'zh-cn') AS 'Chinês  
Simplificado',  
        FORMAT(@d, 'D', 'pt-br') AS 'Português  
Brasileiro'  
GO
```



■ Funções para manipulação de strings

Inglês Americano

Inglês Britânico

Alemão

Sunday, August 26, 2018

26 August 2018

Sonntag, 26. August 2018

Chinês Simplificado

Português Brasileiro

2018年8月26日

domingo, 26 de agosto de 2018



■ Funções para manipulação de strings

-- Exibe o nome do aluno e a data de nascimento

```
SELECT Nome_Aluno AS 'Aluno',  
       Data_Nasc AS 'Data de Nascimento',  
       FORMAT(Data_Nasc, 'D', 'pt-br') AS  
       'Data de Nascimento'  
FROM Alunos  
GO
```



■ Funções para manipulação de strings

Aluno	Data de Nascimento	Data de Nascimento
-----	-----	-----
Maria Cristina	1985-10-12	sábado, 12 de outubro de 1985
Ana Paula Lima	1980-12-01	segunda-feira, 1 de dezembro de 1980
Carlos Renato	1979-05-10	quinta-feira, 10 de maio de 1979
Hugo Silva	1975-10-02	quinta-feira, 2 de outubro de 1975
Marcos Antônio	1985-10-23	quarta-feira, 23 de outubro de 1985
Gislaine Silva	1983-11-06	domingo, 6 de novembro de 1983
.....
Alexandro Duarte	1974-05-29	quarta-feira, 29 de maio de 1974
Maurício dos Santos	1988-04-13	quarta-feira, 13 de abril de 1988
Mary Ann Duarte	1989-04-13	quinta-feira, 13 de abril de 1989
Gabriela Pereira	1983-07-29	sexta-feira, 29 de julho de 1983
André César	1984-12-30	domingo, 30 de dezembro de 1984
Edson Lopes	1979-01-11	quinta-feira, 11 de janeiro de 1979

(50 row(s) affected)



■ Funções para manipulação de strings

Função	Descrição
<code>CONCAT (expressão_1, expressão_2, ...)</code>	Retorna uma cadeia de caracteres que é o resultado da concatenação de dois ou mais valores.
<code>STUFF (expressão_original, início, tamanho, expressão_inserção)</code>	Insere uma cadeia de caracteres em outra cadeia de caracteres. Ela exclui um tamanho especificado de caracteres da primeira cadeia na posição inicial e, em seguida, insere a segunda cadeia na primeira, na posição inicial.
<code>REVERSE (expressão)</code>	Retorna a ordem inversa de um valor da cadeia de caracteres.
<code>REPLICATE (expressão, inteiro)</code>	Repete um valor da cadeia de caracteres um número especificado de vezes.



■ Funções para manipulação de strings

```
-- Demonstra a utilização de CONCAT, STUFF, REVERSE e REPLICATE  
SELECT CONCAT('Paulo', 'Giovani') AS 'CONCAT',  
        CONCAT('Rua ', 'João XXIII, ', '15', ' - São  
Paulo') AS 'Endereço',  
        STUFF('Paulo Giovani', 2, 1, 'TEXT0') AS  
'STUFF',  
        REVERSE('Paulo Giovani') AS 'REVERSE',  
        REPLICATE(' * ', 10)      AS 'REPLICATE 1',  
        'Paulo' + REPLICATE('.', 5) + ': ' + '6666-  
6666' AS 'REPLICATE 2'
```

GO



■ Funções para manipulação de strings

CONCAT

Endereço

STUFF

PauloGiovani	Rua João XXIII, 15 - São Paulo	PTEXT0ulo Giovanni
--------------	--------------------------------	--------------------

REVERSE

REPLICATE 1

REPLICATE 2

inavoiG oluaP	* * * * *	Paulo.....: 6666-6666
---------------	-----------	-----------------------



■ Funções para manipulação de strings

-- Exibe a data de nascimento dos alunos

```
SELECT Nome_Aluno AS 'Nome do Aluno',  
       DATEPART(DAY, Data_Nasc) AS 'Dia',  
       DATENAME(DW, Data_Nasc) AS 'Dia da Semana',  
       DATENAME(M, Data_Nasc) AS 'Nome do Mês',  
       DATEPART(YEAR, Data_Nasc) AS 'Ano',  
       CONCAT(DATENAME(DW, Data_Nasc), ', ',  
              DATEPART(DAY, Data_Nasc), ' de ',  
              DATENAME(M, Data_Nasc), ' de ',  
              DATEPART(YEAR, Data_Nasc), '.') AS 'Data de  
Nascimento'  
FROM Alunos
```



■ Funções para manipulação de strings

Nome do Aluno	Dia	Dia da Semana	Nome do Mês	Ano	Data de Nascimento
Maria Cristina	12	Sábado	Outubro	1985	Sábado, 12 de Outubro de 1985.
Ana Paula Lima	1	Segunda-Feira	Dezembro	1980	Segunda-Feira, 1 de Dezembro de 1980.
Carlos Renato	10	Quinta-Feira	Maio	1979	Quinta-Feira, 10 de Maio de 1979.
Hugo Silva	2	Quinta-Feira	Outubro	1975	Quinta-Feira, 2 de Outubro de 1975.
Marcos Antônio	23	Quarta-Feira	Outubro	1985	Quarta-Feira, 23 de Outubro de 1985.
Gislaine Silva	6	Domingo	Novembro	1983	Domingo, 6 de Novembro de 1983.
.....
Mônica Silva	17	Segunda-Feira	Abril	1978	Segunda-Feira, 17 de Abril de 1978.
Marília dos Santos	15	Domingo	Abril	1984	Domingo, 15 de Abril de 1984.
Guilherme dos Santos	11	Quarta-Feira	Outubro	1978	Quarta-Feira, 11 de Outubro de 1978.
Heidi Lima	11	Sexta-Feira	Agosto	1978	Sexta-Feira, 11 de Agosto de 1978.
Amílcar Júnior	10	Sábado	Abril	1976	Sábado, 10 de Abril de 1976.
Alexandro Duarte	29	Quarta-Feira	Maio	1974	Quarta-Feira, 29 de Maio de 1974.
Maurício dos Santos	13	Quarta-Feira	Abril	1988	Quarta-Feira, 13 de Abril de 1988.
Mary Ann Duarte	13	Quinta-Feira	Abril	1989	Quinta-Feira, 13 de Abril de 1989.
Gabriela Pereira	29	Sexta-Feira	Julho	1983	Sexta-Feira, 29 de Julho de 1983.
André César	30	Domingo	Dezembro	1984	Domingo, 30 de Dezembro de 1984.
Edson Lopes	11	Quinta-Feira	Janeiro	1979	Quinta-Feira, 11 de Janeiro de 1979.

(50 row(s) affected)



■ Funções para manipulação de strings

Função	Descrição
<code>SUBSTRING (expressão, início, tamanho)</code>	Retorna parte de uma expressão de caractere, binária, de texto ou de imagem (a partir do SQL Server 2008).
<code>LEFT (expressão, tamanho)</code>	Retorna a parte da esquerda de uma cadeia de caracteres com o número de caracteres especificado.
<code>RIGHT (expressão, tamanho)</code>	Retorna a parte da direita de uma cadeia de caracteres com o número de caracteres especificado.
<code>LOWER (expressão)</code>	Converte em minúsculas uma expressão de caracteres.
<code>UPPER (expressão)</code>	Converte em maiúsculas uma expressão de caracteres.



■ Funções para manipulação de strings

Função	Descrição
<code>LTRIM(expressão)</code>	Retorna uma cadeia de caracteres após remover todos os espaços em branco à esquerda.
<code>RTRIM(expressão)</code>	Retorna uma cadeia de caracteres após remover todos os espaços em branco à direita.



■ Funções para manipulação de strings

-- Demonstra o uso de SUBSTRING, LEFT, RIGHT, LOWER, UPPER, LTRIM e RTRIM

```
SELECT Nome_Aluno          AS 'Nome do aluno',  
       SUBSTRING(Nome_Aluno, 1, 1) AS 'Inicial',  
       LOWER(Nome_Aluno) AS 'Nome em minúsculas',  
       UPPER(Nome_Aluno) AS 'Nome em maiúsculas',  
       LEN(Nome_Aluno)    AS 'Qtd caracteres',  
       LEFT(Nome_Aluno, 3) AS 'LEFT',  
       RIGHT(Nome_Aluno, 3) AS 'RIGHT',  
       RIGHT(RTRIM(Nome_Aluno), 3) AS 'RTRIM',  
       '      ' + Nome_Aluno      AS 'LTRIM 1',  
       LTRIM('      ' + Nome_Aluno) AS 'LTRIM 2'  
  
FROM Alunos
```

GO



■ Funções para manipulação de strings

Nome do aluno	Inicial	Nome em minúsculas	Nome em maiúsculas
-----	-----	-----	-----
Maria Cristina	M	maria cristina	MARIA CRISTINA
Ana Paula Lima	A	ana paula lima	ANA PAULA LIMA
Carlos Renato	C	carlos renato	CARLOS RENATO
.....
André César	A	andré césar	ANDRÉ CÉSAR
Edson Lopes	E	edson lopes	EDSON LOPES

Qtd caracteres	LEFT	RIGHT	RTRIM	LTRIM 1	LTRIM 2
-----	-----	-----	-----	-----	-----
14	Mar		ina	Maria Cristina	Maria Cristina
14	Ana		ima	Ana Paula Lima	Ana Paula Lima
13	Car		ato	Carlos Renato	Carlos Renato
.....
11	And		sar	André César	André César
11	Eds		pes	Edson Lopes	Edson Lopes



■ Funções para conversão de dados

Função	Descrição
<code>CAST (expressão AS tipo_dados [(tamanho)])</code>	Converte uma expressão de um tipo de dados para outro (ANSI).
<code>CONVERT (tipo_dados [(tamanho)], expressão [, estilo])</code>	Converte uma expressão de um tipo de dados para outro (SQL Server). Permite a utilização de um estilo para formatar o resultado.
<code>PARSE (expressão AS tipo_dados [USING cultura])</code>	Retorna o resultado de uma expressão, convertida no tipo de dados solicitado no SQL Server. É utilizado para converter cadeias de caracteres em formatos de data/hora ou numéricos



■ Funções para conversão de dados

Função

Descrição

```
TRY_CAST(expressão AS  
tipo_dados  
[(tamanho)])
```

Retorna uma conversão de valor ao tipo de dados especificado se a conversão for bem-sucedida. Caso contrário, retorna nulo.

```
TRY_CONVERT(tipo_dado  
s [(tamanho)],  
expressão [, estilo])
```

Retorna uma conversão de valor ao tipo de dados especificado se a conversão for bem-sucedida. Caso contrário, retorna nulo.

```
TRY_PARSE(expressão  
AS tipo_dados [USING  
cultura])
```

Retorna uma conversão de valor ao tipo de dados especificado se a conversão for bem-sucedida. Caso contrário, retorna nulo. É utilizado para converter cadeias de caracteres em formatos de data/hora ou numéricos.



■ Utilizando CAST e CONVERT

-- Exemplo básico de utilização de CAST e CONVERT

```
DECLARE @valor AS DECIMAL(5, 2) = 156.90
```

```
SELECT  CAST(@valor AS CHAR(20))           AS 'CAST',  
        CONVERT(DECIMAL(10,5), @valor) AS 'CONVERT'
```

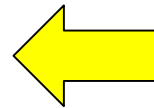
```
GO
```

CAST

CONVERT

156.90

156.90000



(1 row(s) affected)



■ Utilizando CAST e CONVERT

-- Exemplo de conversão de um inteiro com CAST e CONVERT

```
DECLARE @valor AS INT = 100
```

```
SELECT CAST(@valor AS CHAR(20))           AS 'CAST  
1',
```

```
      CAST(@valor AS DECIMAL(10, 5)) AS 'CAST  
2',
```

```
      CONVERT(CHAR(20), @valor)           AS 'CONVERT  
1',
```

```
      CONVERT(DECIMAL(10,5), @valor) AS 'CONVERT  
2'
```



■ Utilizando CAST e CONVERT com datas

-- Conversão de uma data utilizando CAST e CONVERT

```
DECLARE @data AS DATE = '01/20/2017'
```

```
SELECT  CAST(@data AS CHAR(20))      AS 'CAST (Padrão)',  
        CONVERT(CHAR(20), @data)    AS 'CONVERT  
(Padrão)',  
        CONVERT(CHAR(20), @data, 101) AS 'EUA  
(mm/dd/aaaa)',  
        CONVERT(CHAR(20), @data, 103) AS 'Brasil  
(dd/mm/aaaa)',  
        CONVERT(CHAR(20), @data, 111) AS 'Japão  
(aaaa/mm/dd)'
```



GO



■ Utilizando CAST e CONVERT com datas

CAST (Padrão)	CONVERT (Padrão)	EUA (mm/dd/aaaa)
-----	-----	-----
2017-01-20	2017-01-20	01/20/2017
		↑
Brasil (dd/mm/aaaa)	Japão (aaaa/mm/dd)	
-----	-----	
20/01/2017	2017/01/20	
↑	↑	



■ Utilizando PARSE

```
-- Demonstra a utilização do PARSE para converter data e dinheiro  
SELECT PARSE('Segunda-feira, 05 junho 2017' AS DATE USING  
'pt-BR') AS 'Data Brasileira',  
        PARSE('Monday, 05 June 2017' AS DATE USING 'en-  
US') AS 'Data Americana',  
        PARSE('R$ 345,98' AS MONEY USING 'pt-BR') AS  
'Dinheiro Brasileiro',  
        PARSE('$ 345.98' AS MONEY USING 'en-US') AS  
'Dinheiro Americano',  
        PARSE('¥ 345.98' AS MONEY USING 'jp-JP') AS 'Dinheiro  
Japonês',  
        PARSE('06/05/2017' AS DATETIME2) AS 'Data Padrão'
```



■ Utilizando PARSE

Data Brasileira Data Americana Dinheiro Brasileiro

2017-06-05

2017-06-05

345,98



Dinheiro Americano

Dinheiro Japonês

Data Padrão

345,98

345,98

2017-06-05 00:00:00.0000000



- Utilizando TRY_CAST, TRY_CONVERT e TRY_PARSE

-- Exemplo de validação utilizando TRY_CAST,
-- TRY_CONVERT e TRY_PARSE

```
DECLARE @data AS CHAR(10) = '20/01/2017'
```

```
SELECT TRY_CAST(@data AS DATE) AS 'TRY CAST',  
       TRY_CONVERT(DATE, @data, 103) AS  
'Brasil (dd/mm/aaaa)',  
       TRY_CONVERT(DATE, @data, 101) AS 'EUA  
(mm/dd/aaaa)',  
       TRY_PARSE(@data AS DATE) AS 'TRY_PARSE'
```

GO



- Utilizando TRY_CAST, TRY_CONVERT e TRY_PARSE

TRY CAST	Brasil (dd/mm/aaaa)	EUA (mm/dd/aaaa)	TRY_PARSE
-----	-----	-----	-----
NULL	2017-01-20	NULL	NULL

(1 row(s) affected)



- **Na próxima aula veremos**
 - Sequências e SQL procedural.

