



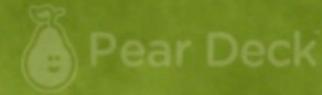
RDB & SQL

Session 3



I've completed the pre-class content?

True



Pear Deck

False

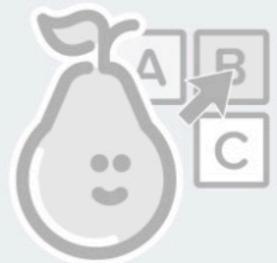


Pear Deck



Students choose an option

Pear Deck Interactive Slide
Do not remove this bar



No Multiple Choice Response
You didn't answer this question

NEXT SLIDE



Table of Contents

- ▶ Date and Time Data Types
- ▶ Return Date or Time Parts
- ▶ Return Date and Time Difference Values
- ▶ Modify Date and Time Values
- ▶ Validate Date and Time Values



Date Functions

► Date and Time Data Types



Data Types	Format
time	hh:mm:ss[.nnnnnnn]
date	YYYY-MM-DD
smalldatetime	YYYY-MM-DD hh:mm:ss
datetime	YYYY-MM-DD hh:mm:ss[.nnn]
datetime2	YYYY-MM-DD hh:mm:ss[.nnnnnnnn]
datetimeoffset	YYYY-MM-DD hh:mm:ss[.nnnnnnnn] [+ -]hh:mm

► GETDATE() Function



Function

GETDATE()

Description

Returns the current system date and time of the operating system on which the SQL Server is running.

Return Data Type

datetime





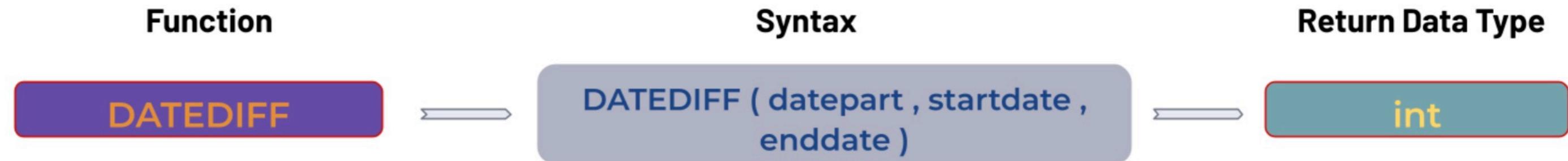
► Return Date or Time Parts



Function	Syntax	Return Data Type
DATENAME	DATENAME (datepart , date)	nvarchar
DATEPART	DATEPART (datepart , date)	int
DAY	DAY (date)	int
MONTH	MONTH (date)	int
YEAR	YEAR (date)	int



► Return Date and Time Difference Values





► Modify Date and Time Values



Function

Syntax

Return Data Type

DATEADD

DATEADD (datepart , number , date)

The data type of
the date
argument

EOMONTH

EOMONTH (start_date [, month_to_add])

Return type is the
type of the
start_date
argument, or
alternately, the
date data type.



► Validate Date and Time Values



Function	Syntax	Return Data Type
----------	--------	------------------

ISDATE

ISDATE (expression)

int(boolean)





Query Time For You

Question: Write a query returns orders that are shipped more than two days after the order date.

Expected Output:



	order_id	customer_id	order_status	order_date	required_date	shipped_date	store_id	staff_id	DATE_DIFF
1	5	1324	4	2018-01-03	2018-01-06	2018-01-06	2	6	3
2	9	60	4	2018-01-05	2018-01-08	2018-01-08	1	2	3
3	12	91	4	2018-01-06	2018-01-08	2018-01-09	1	2	3
4	13	873	4	2018-01-08	2018-01-11	2018-01-11	2	6	3
5	14	258	4	2018-01-09	2018-01-11	2018-01-12	1	3	3
6	15	450	4	2018-01-09	2018-01-10	2018-01-12	2	7	3
7	16	552	4	2018-01-12	2018-01-15	2018-01-15	1	3	3
8	20	923	4	2018-01-14	2018-01-16	2018-01-17	1	2	3



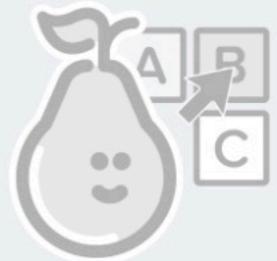
▶ String Functions



I've completed the pre-class content?



Pear Deck Interactive Slide
Do not remove this bar



No Multiple Choice Response
You didn't answer this question



Table of Contents

- ▶ Character Strings Data Types
- ▶ String Functions
- ▶ Related Topics

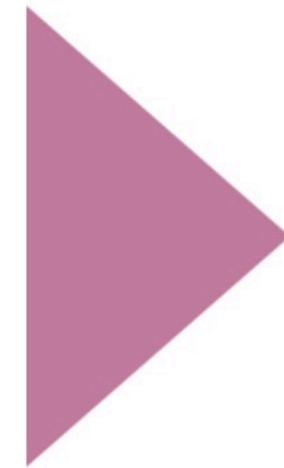
Character Strings Data Types



Data Type	Lower Limit	Upper Limit	Memory
char	0 chars	8000 chars	n bytes
varchar	0 chars	8000 chars	n bytes + 2 bytes
text	0 chars	2.147.483.647 chars	n bytes + 4 bytes
nchar	0 chars	4000 chars	2 times n bytes
nvarchar	0 chars	4000 chars	2 times n bytes + 2 bytes
ntext	0 chars	1.073.741.823 chars	2 times the string length



ntext, text data types will be removed in a future version of SQL Server. Avoid using these data types in new development work, and plan to modify applications that currently use them. Use **nvarchar(max), varchar(max)** instead.



String Functions

► LEN(), CHARINDEX(), PATINDEX()



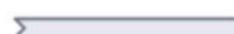
Function Syntax

`LEN (input_string)`



Return number of characters of a character string.
(Excluding spaces)

`CHARINDEX (substring,
string [, start_location])`



Returns the position of a substring within a
specified string

`PATINDEX ('%pattern%',
input_string)`



Returns the starting position of the first occurrence
of a pattern in a string.

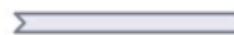


► LEFT(), RIGHT(), SUBSTRING()



Function Syntax

```
LEFT( input_string ,  
      number_of_characters )
```



Description

Extract a given a number of characters from a character string starting from the left

```
RIGHT ( input_string ,  
        number_of_characters )
```



Extract a given a number of characters from a character string starting from the right

```
SUBSTRING( input_string ,  
            start , length )
```



Extract a substring within a string starting from a specified location with a specified length



► LOWER(), UPPER(), STRING_SPLIT()



Function Syntax

LOWER (input_string)



Convert a string to lowercase

UPPER (input_string)



Convert a string to uppercase

STRING_SPLIT (input_string
, separator)



A table-valued function that splits a string into rows
of substrings based on a specified separator.

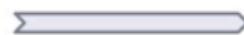


► TRIM(), LTRIM(), RTRIM()



Function Syntax

TRIM ([removed_characters
, from] input_string)



Return a new string from a specified string after
removing all leading and trailing blanks or
characters

LTRIM(input_string)



Return a new string from a specified string after
removing all leading blanks

RTRIM (input_string)



Return a new string from a specified string after
removing all trailing blanks

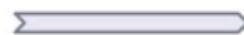


► REPLACE(), STR()



Function Syntax

```
REPLACE ( input_string ,  
          substring , new_substring )
```



Description

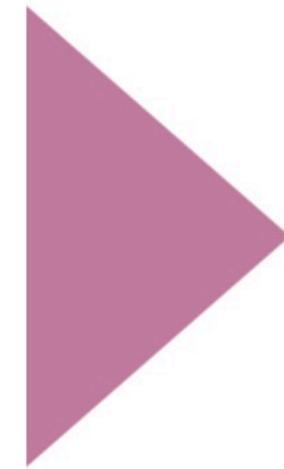
Replace all occurrences of a substring, within a string, with another substring

```
STR ( float_expression [ ,  
           length [ , decimal ] ] )
```



Returns character data converted from numeric data.





**Related Topics
(CAST , CONVERT,
COALESCE, NULLIF,
ROUND)**



► CAST(), CONVERT()

Function Syntax

```
CAST ( expression AS  
target_type [ ( length ) ] )
```



Cast a value of one type to another.

```
CONVERT ( target_type [ (  
length ) ] , expression [ , style  
])
```



Convert a value of one type to another.

► SQL Server Datetime Style Codes



Style Code	Style	Format	Example
0 or 100	Default. Equivalent to not specifying a style code.	mon dd yyyy hh:mmAM	Sep 8 2007 9:00PM
1	USA date.	mm/dd/yy	9.08.2007
2	ANSI date.	yy.mm.dd	7.09.2008
3	UK / French date.	dd/mm/yy	8.09.2007
4	German date.	dd.mm.yy	8.09.2007
5	Italian date.	dd-mm-yy	8.09.2007
6	Abbreviated month.	dd mmm yy	08 Sep 07
7	Abbreviated month.	mmm dd, yy	Sep 08, 07
8 or 108	24 hour time.	HH:mm:ss	21:00:00
9 or 109	Default formatting with seconds and milliseconds appended.	mon dd yyyy hh:mm:ss:fffAM	Sep 8 2007 9:00:00:000PM
10	USA date with hyphen separators.	mm-dd-yy	9.08.2007
11	Japanese date.	yy/mm/dd	7.09.2008
12	ISO date.	yymmdd	70908
13 or 113	European default with seconds and milliseconds.	dd mon yyyy HH:mm:ss:fff	08 Sep 2007 21:00:00:000
14 or 114	24 hour time with milliseconds.	HH:mm:ss:fff	21:00:00:000
20 or 120	ODBC canonical date and time.	yyyy-mm-dd HH:mm:ss	8.09.2007 21:00
21 or 121	ODBC canonical date and time with milliseconds.	yyyy-mm-dd HH:mm:ss.fff	2007-09-08 21:00:00.000
101	USA date with century.	mm/dd/yyyy	9.08.2007
102	ANSI date with century.	yyyy.mm.dd	8.09.2007
103	UK / French date with century.	dd/mm/yyyy	8.09.2007
104	German date with century.	dd.mm.yyyy	8.09.2007
105	Italian date with century.	dd-mm-yyyy	8.09.2007
106	Abbreviated month with century.	dd mmm yyyy	08 Sep 2007
107	Abbreviated month with century.	mmm dd, yyyy	Sep 08, 2007
110	USA date with hyphen separators and century.	mm-dd-yyyy	9.08.2007
111	Japanese date with century.	yyyy/mm/dd	8.09.2007
112	ISO date with century.	yymmdd	20070908
126	ISO8601, for use in XML.	yyy-mm-ddThh:mm:ss	2007-09-08T21:00:00

► SQL Server Datetime Formatting



Converting a Datetime to a Varchar

```
SELECT CONVERT(VARCHAR, GETDATE(), 6)
```

Converting a Varchar to a Date

```
SELECT convert(DATE, '25 Oct 21', 6)
```

► ROUND(), ISNULL()



Function Syntax

ROUND(number, decimals,
[operation])



Rounds a number to a specified number of decimal places.

ISNULL(check expression,
replacement value)



Replaces NULL with the specified replacement value.

► COALESCE, NULLIF, ISNUMERIC



Function Syntax

Description

`COALESCE(Expression1, [E2, ... , En])`



Returns the first non-null argument.

`NULLIF(Expression1, Expression2)`



Returns NULL if two arguments are equal.
Otherwise, it returns the first expression.

`ISNUMERIC(Expression)`



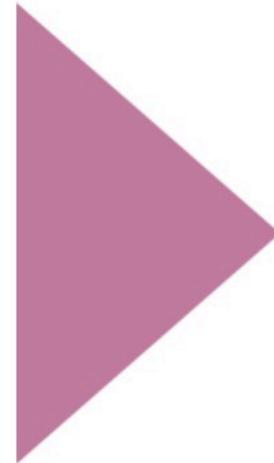
Determines whether an expression is a valid numeric type.





Query Time For You

Question: How many customers have yahoo mail?



Expected Output: **207**



Query Time For You

Question: Write a query that returns the characters before the '@' character in the email column.

Expected Output:

	email	Chars
1	DianeFlosi@msn.com	DianeFlosi
2	NanaGaines@msn.com	NanaGaines
3	TeddyMills@hotmail.com	TeddyMills
4	NicolasGarrison@aliexpress.com	NicolasGarrison
5	TessieMullen@aol.com	TessieMullen
6	CyrilKohl@msn.com	CyrilKohl
7	WilliamWilliams@gmail.com	WilliamWilliams
8	ElizabethLevy@yahoo.com	ElizabethLevy
9	LindaMontezuma@gmail.com	LindaMontezuma
10	MelissaPaulas@gmail.com	MelissaPaulas
11	StephaineChapman@hotmail.c...	StephaineChap...
12	DarenRollins@aol.com	DarenRollins
...



Query Time For You

Question: Add a new column to the customers table that contains the customers' contact information.
If the phone is not null, the phone information will be printed, if not, the email information will be printed.

Expected Output:

The screenshot shows a SQL query results window with two tabs: 'Results' and 'Messages'. The 'Results' tab displays a table with 10 rows of customer data. The columns are: customer_id, first_name, last_name, phone, email, and contact. The 'contact' column contains either the phone number or the email address based on the nullability of the 'phone' column. The 'Messages' tab is empty.

	customer_id	first_name	last_name	phone	email	contact
1	1	Diane	Flosi	+1-28 - 022 - 0227	DianeFlosi@msn.com	+1-28 - 022 - 0227
2	2	Nana	Gaines	+1-20 - 431 - 4316	NanaGaines@msn.com	+1-20 - 431 - 4316
3	3	Teddy	Mills	NULL	TeddyMills@hotmail.com	TeddyMills@hotmail.com
4	4	Nicolas	Garrison	+1-11 - 675 - 6753	NicolasGarrison@aliex...	+1-11 - 675 - 6753
5	5	Tessie	Mullen	+1-90 - 740 - 7404	TessieMullen@aol.com	+1-90 - 740 - 7404
6	6	Cyril	Kohl	+1-40 - 658 - 6584	CyrilKohl@msn.com	+1-40 - 658 - 6584
7	7	William	Williams	+1-35 - 042 - 0427	WilliamWilliams@gmai...	+1-35 - 042 - 0427
8	8	Elizabeth	Levy	+1-75 - 977 - 9779	ElizabethLevy@yahoo....	+1-75 - 977 - 9779
9	9	Linda	Montezuma	+1-48 - 001 - 0019	LindaMontezuma@gm...	+1-48 - 001 - 0019

Query executed successfully. (local) (15.0 RTM) DESKTOP-3E95HEO\Datas... SampleRetail 00:00:00 2.000 rows



Query Time For You

Question: Write a query that returns the name of the streets, where the third character of the streets is numeric.

Expected Output:

Screenshot of a SQL query results window showing the expected output:

	street	third_char
1	69734 E Carrillo St	7
2	128 Bransten Rd	8
3	6916 W Main St	1
4	3958 S Dupont Hwy #7	5
5	83649 W Belmont Ave	6
6	2215 Prosperity Dr	1
7	13252 Lighthouse Ave	2
8	810 N La Brea Ave	0
9	393 Hammond Dr	3
10	70205 Pioneer Ct	2

(local) (15.0 RTM) | DESKTOP-3E95HEO\DataSc... | SampleRetail | 00:00:00 | 1.170 rows



Query Time

Question: Split the mail addresses into two parts from '@', and place them in separate columns.

Expected Output:

	email	Left_part	Right_part
1	DianeFlosi@msn.com	DianeFlosi	msn.com
2	NanaGaines@msn.com	NanaGaines	msn.com
3	TeddyMills@hotmail.com	TeddyMills	hotmail.com
4	NicolasGarrison@aliexpress.com	NicolasGarrison	aliexpress.com
5	TessieMullen@aol.com	TessieMullen	aol.com
6	CyrilKohl@msn.com	CyrilKohl	msn.com
7	WilliamWilliams@gmail.com	WilliamWilliams	gmail.com
8	ElizabethLevy@yahoo.com	ElizabethLevy	yahoo.com
9	LindaMontezuma@gmail.com	LindaMontezu...	gmail.com
10	MelissaPaulas@gmail.com	MelissaPaulas	gmail.com

Query executed successfully.

(local) (15.0 RTM) DESKTOP-3E95HEO\Datas... SampleRetail 00:00:00 | 2.000 rows

Query Time



Question: The street column has some string characters (5C, 43E, 234F, etc.) that are mistakenly added to the end of the numeric characters in the first part of the street records. Remove these typos in this column.

Expected Output:

The screenshot shows a SQL query results window with two columns: 'street' and 'new_street'. The 'street' column contains original street names with typos, and the 'new_street' column shows the corrected versions. The typos are removed from the numeric parts of the street names.

	street	new_street
1	5C Tomahawk Dr	5 Tomahawk Dr
2	8C County Center Dr #647	8 County Center Dr #647
3	4C Otis St	4 Otis St
4	9C N College Ave #3	9 N College Ave #3
5	9C Hwy	9 Hwy
6	3C Secor Rd	3 Secor Rd

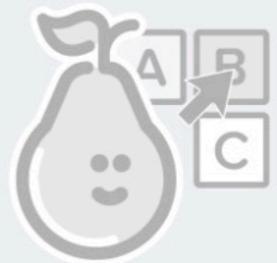
Query executed successfully | (local) (15.0 RTM) | DESKTOP-3E95HEO\Datas... | SampleRetail | 00:00:00 | 26 rows

Is everything clear so far?



Students choose an option

Pear Deck Interactive Slide
Do not remove this bar



No Multiple Choice Response
You didn't answer this question

How well did you like this lesson?

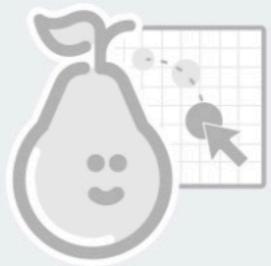


Pear Deck

Pear Deck Interactive Slide
Do not remove this bar



Students, drag the icon!



No Draggable™ Response
You didn't answer this question