**Script Constraint**

Script do tópico.

1. CREATE DATABASE BD\_CONSTRAINTS
3. USE BD\_CONSTRAINTS;
5. /\*IDENTITY\*/
7. CREATE TABLE tb\_identity(
8. Cod INT IDENTITY PRIMARY KEY,
9. NOME VARCHAR(50)
10. )
12. INSERT INTO tb\_identity (NOME)
13. VALUES ('LINHA 1'), ('LINHA 2')
15. -- APRESENTAR ERRO
16. INSERT INTO tb\_identity (Cod,NOME)
17. VALUES (3,'LINHA 3')
19. SELECT \* FROM tb\_identity
21. /\*PK\*/
23. CREATE TABLE tb\_pk (
24. Cod INT CONSTRAINT pk\_test PRIMARY KEY,
25. Nome VARCHAR(50)
26. )
28. INSERT INTO tb\_pk (Cod,NOME)
29. VALUES (1,'LINHA 1'),(2,'LINHA 3')
31. -- APRESENTAR ERRO
32. INSERT INTO tb\_pk (Cod,NOME)
33. VALUES (2,'LINHA 4')

36. SELECT \* FROM tb\_pk
38. /\*FK\*/
40. CREATE TABLE tb\_fk\_marca (
41. Cod INT PRIMARY KEY IDENTITY,
42. Nome VARCHAR(50)
43. )
45. CREATE TABLE tb\_fk\_carro(
46. Placa varchar(6) PRIMARY KEY,
47. modelo VARCHAR(50),
48. fk\_marca INT FOREIGN KEY REFERENCES tb\_fk\_marca(Cod)
49. )

52. INSERT INTO tb\_fk\_marca (NOME)
53. VALUES ('VW'), ('CHEVROLET'), ('TOYOTA')

56. INSERT INTO tb\_fk\_carro (Placa,modelo,fk\_marca)
57. VALUES ('PLA001', 'FOX', 1)
58. ,('PLA002', 'ONIX SEDAN', 2)

61. -- APRESENTAR ERRO
62. INSERT INTO tb\_fk\_carro (Placa,modelo,fk\_marca)
63. VALUES ('PLA004', 'ARGO', 4)
65. SELECT \* FROM tb\_fk\_marca
66. SELECT \* FROM tb\_fk\_carro
68. /\*UNIQUE\*/
70. CREATE TABLE tb\_unique(
71. Cod INT PRIMARY KEY IDENTITY,
72. Nome VARCHAR(50),
73. CPF CHAR(11) CONSTRAINT UQ\_CPF UNIQUE
74. )
76. INSERT INTO tb\_unique (NOME, CPF)
77. VALUES ('PESSOA A', '11122233344'), ('PESSOA B', '11122233345')
79. --APRESENTAR ERRO
80. INSERT INTO tb\_unique (NOME, CPF)
81. VALUES ('PESSOA C', '11122233344')
83. SELECT \* FROM tb\_unique
85. /\*NOT NULL\*/
87. CREATE TABLE tb\_notnull(
88. Cod INT PRIMARY KEY IDENTITY,
89. NOME VARCHAR(50) NOT NULL,
90. DT DATE NULL
91. )

94. INSERT INTO tb\_notnull ( NOME, DT)
95. VALUES ('A', GETDATE())
97. --APRESENTAR ERRO
98. INSERT INTO tb\_notnull ( NOME, DT)
99. VALUES (NULL, GETDATE())
101. SELECT \* FROM tb\_notnull
103. /\*DEFAULT\*/
105. CREATE TABLE tb\_default(
106. Cod INT PRIMARY KEY IDENTITY,
107. Nome VARCHAR(50),
108. dtInsert DATE CONSTRAINT DF\_DATA DEFAULT (GETDATE())
109. )
111. INSERT INTO tb\_default ( NOME, dtInsert)
112. VALUES ('A', '2020-04-15')
114. --VERIFICAR VALOR DEFAULT
115. INSERT INTO tb\_default (NOME)
116. VALUES ('B')
118. SELECT \* FROM tb\_default
120. /\*CHECK\*/

123. CREATE TABLE tb\_check(
124. Cod INT PRIMARY KEY IDENTITY,
125. NOME VARCHAR(50) NOT NULL,
126. VALOR NUMERIC (6,2) CHECK (VALOR > 0),
127. LETRA CHAR(1) CHECK (LETRA IN ('A', 'B', 'C'))
128. )
130. INSERT INTO tb\_check ( NOME, VALOR, LETRA)
131. VALUES ('linha 1', 100.99, 'A')
133. --APRESENTAR ERRRO 1: CAMPO VALOR
134. INSERT INTO tb\_check ( NOME, VALOR, LETRA)
135. VALUES ('linha 2', 0, 'A')
137. --APRESENTAR ERRRO 2 : CAMPO LETRA
138. INSERT INTO tb\_check ( NOME, VALOR, LETRA)
139. VALUES ('linha 2', 20, 'D')
141. SELECT \* FROM tb\_check

**Script**

SCRIPT DO TÓPICO

1. CREATE DATABASE BD\_CONSULTAS;
3. USE BD\_CONSULTAS;
5. CREATE TABLE TB\_FUNC (
6. MATRICULA INT PRIMARY KEY IDENTITY,
7. NOME VARCHAR (255) NOT NULL,
8. DATANASCIMENTO DATE NOT NULL,
9. SALARIO MONEY NULL,
10. DATAADMISSAO DATE DEFAULT (GETDATE()),
11. DATADEMISSAO DATE NULL,
12. INICIOFERIAS DATE NULL,
13. FIMFERIAS DATE NULL,
14. STATUS\_FUNC VARCHAR (20) CHECK (STATUS\_FUNC IN ('ATIVO','INATIVO','FERIAS','LICENÇA','INSS')),
15. CODDEPTO INT)
17. INSERT INTO TB\_FUNC (NOME, DATANASCIMENTO,SALARIO,DATAADMISSAO,DATADEMISSAO,INICIOFERIAS,FIMFERIAS, STATUS\_FUNC, CODDEPTO)
18. VALUES ('ANA MARIA', '2000-01-01', 1200.55, '2018-05-01',NULL,NULL,NULL, 'ATIVO', 1),
19. ('JOSE HENRIQUE', '1998-11-20', 2575.55, '2005-09-01','2017-12-01',NULL,NULL, 'INATIVO', 7),
20. ('ANA MARIA', '2002-08-21', 950.00, '2019-01-01',NULL,NULL,NULL, 'ATIVO', 6),
21. ('LUAN FELIX', '1991-09-28', 3500.00, '2013-04-01',NULL,NULL,NULL, 'ATIVO', 1),
22. ('FELIPE JOSE DOS SANTOS', '1996-01-11', 4000, '2011-05-01','2015-01-29',NULL,NULL, 'INATIVO', 2),
23. ('MARCELO JOSE', '1980-10-05', 7000, '2000-05-01',NULL,'2019-05-01','2019-06-01', 'ATIVO', 2),
24. ('MARIANA MARIA', '1987-02-08', 4500, '2010-01-01',NULL,NULL,NULL, 'INSS', 3),
25. ('JULIANA MARIA DOS SANTOS', '2002-01-01', 2000, '2017-05-01',NULL,NULL,NULL, 'LICENÇA', 5),
26. ('MARIA ALICIA', '2001-01-01', 950, '2018-05-01',NULL,NULL,NULL, 'ATIVO', 4)
28. INSERT INTO TB\_FUNC (NOME, DATANASCIMENTO,SALARIO,STATUS\_FUNC, CODDEPTO)
29. VALUES ('MARIA ALICIA', '2003-09-18', 950, 'ATIVO', 3)
31. /\*order by\*/
33. -- asc
34. SELECT \*
35. FROM TB\_FUNC
36. ORDER BY NOME
38. -- por 2 colunas asc
39. SELECT \*
40. FROM TB\_FUNC
41. ORDER BY NOME , SALARIO
43. -- desc
44. SELECT \*
45. FROM TB\_FUNC
46. ORDER BY NOME DESC
48. -- por 2 colunas, 1 asc, 1 desc
49. SELECT \*
50. FROM TB\_FUNC
51. ORDER BY NOME ASC, SALARIO DESC
53. /\* OPERADORES LÓGICOS: AND | OR | NOT \*/
55. --AND
56. SELECT MATRICULA, NOME, DATADEMISSAO, STATUS\_FUNC
57. FROM TB\_FUNC
58. WHERE DATADEMISSAO IS NULL
59. AND STATUS\_FUNC = 'ATIVO'
61. -- OR
62. SELECT MATRICULA, NOME, DATADEMISSAO, STATUS\_FUNC
63. FROM TB\_FUNC
64. WHERE DATADEMISSAO IS NULL
65. OR STATUS\_FUNC = 'ATIVO'
67. --NOT
68. SELECT
69. MATRICULA, NOME, SALARIO, STATUS\_FUNC
70. FROM TB\_FUNC
71. WHERE NOT STATUS\_FUNC = 'ATIVO'
73. /\* OPERADORES ARITMÉTICOS \*/
75. SELECT
76. NOME,
77. SALARIO + 50 AS ADICAO,
78. SALARIO - 20 AS SUBTRACAO,
79. SALARIO \* 1.20 AS AJUSTE\_20\_PCENTO,
80. SALARIO / 2 AS DIVISAO
81. FROM TB\_FUNC
82. WHERE STATUS\_FUNC = 'ATIVO'
84. /\* OPERADORES RELACIONAIS \*/
86. SELECT \*
87. FROM TB\_FUNC
88. WHERE STATUS\_FUNC = 'ATIVO'

91. SELECT \*
92. FROM TB\_FUNC
93. WHERE DATANASCIMENTO >= '1999-01-01'
95. SELECT \*
96. FROM TB\_FUNC
97. WHERE SALARIO >= 2000
98. AND SALARIO <= 4000
100. SELECT \*
101. FROM TB\_FUNC
102. WHERE STATUS\_FUNC <> 'ATIVO'
104. /\* OPERADORES AUXILIARES \*/
106. -- IS NULL
107. SELECT \*
108. FROM TB\_FUNC
109. WHERE DATADEMISSAO IS NULL
111. -- IS NOT NULL
112. SELECT \*
113. FROM TB\_FUNC
114. WHERE DATADEMISSAO IS NOT NULL
116. -- BETWEEN
118. SELECT \*
119. FROM TB\_FUNC
120. WHERE DATAADMISSAO BETWEEN '2012-01-01' AND '2018-12-20'
122. SELECT \*
123. FROM TB\_FUNC
124. WHERE DATAADMISSAO NOT BETWEEN '2017-01-01' AND '2017-12-20'
126. SELECT \*
127. FROM TB\_FUNC
128. WHERE CODDEPTO BETWEEN 5 AND 15
130. -- IN
131. SELECT \*
132. FROM TB\_FUNC
133. WHERE STATUS\_FUNC IN ('ATIVO','FERIAS')
135. SELECT \*
136. FROM TB\_FUNC
137. WHERE MONTH(DATAADMISSAO) IN (1,2)
139. SELECT \*
140. FROM TB\_FUNC
141. WHERE MONTH(DATAADMISSAO) NOT IN (10,11)
143. --LIKE
144. SELECT \*
145. FROM TB\_FUNC
146. WHERE NOME LIKE 'JOSE'
148. SELECT \*
149. FROM TB\_FUNC
150. WHERE NOME LIKE 'JOSE%'
152. SELECT \*
153. FROM TB\_FUNC
154. WHERE NOME LIKE '%JOSE'
156. SELECT \*
157. FROM TB\_FUNC
158. WHERE NOME LIKE '%JOSE%'
160. /\*CLAUSULAS EXTRAS\*/
162. -- ALIAS
163. SELECT
164. F.NOME AS 'NOME FUNCIONÁRIO',
165. 'DATA DE ADMISSÃO' = F.DATAADMISSAO,
166. F.SALARIO 'SALÁRIO FUNCIONÁRIO'
167. FROM TB\_FUNC AS F
169. --CASE
170. SELECT NOME, STATUS\_FUNC,
171. CASE STATUS\_FUNC
172. WHEN 'ATIVO' THEN 'A'
173. WHEN 'INATIVO' THEN 'I'
174. WHEN 'FERIAS' THEN 'F'
175. ELSE
176. 'O'
177. END SIGLA\_STATUS
178. FROM TB\_FUNC
180. SELECT NOME, SALARIO,
181. CASE
182. WHEN SALARIO > 1200 AND SALARIO < 1500 THEN SALARIO \* 1.1
183. WHEN SALARIO > 1501 AND SALARIO < 2000 THEN SALARIO \* 1.2
184. WHEN SALARIO < 1000 THEN SALARIO
185. END SALARIO\_COM\_REAJUSTE
187. FROM TB\_FUNC
189. -- DISTINCT
191. SELECT DISTINCT NOME
192. FROM TB\_FUNC
194. SELECT DISTINCT NOME, SALARIO
195. FROM TB\_FUNC
197. SELECT DISTINCT NOME, SALARIO, DATANASCIMENTO
198. FROM TB\_FUNC
200. -- TOP
201. SELECT
202. TOP 5 MATRICULA,
203. NOME
204. FROM TB\_FUNC
206. SELECT
207. TOP 5 MATRICULA,
208. NOME
209. FROM TB\_FUNC
210. ORDER BY MATRICULA DESC

**SCRIPT PARA PRATICAR JOINS**

EXECUTAR SCRIPT NO SEU SQL SERVER

1. CREATE DATABASE BD\_JOINS
2. GO
4. USE BD\_JOINS
5. GO
7. CREATE TABLE DEPARTAMENTO (
8. CODIGO INT PRIMARY KEY IDENTITY,
9. DESCRICAO VARCHAR(100) NOT NULL);
10. GO
12. CREATE TABLE FUNCAO (
13. CODIGO INT PRIMARY KEY IDENTITY,
14. DESCRICAO VARCHAR(100))
15. GO
17. CREATE TABLE FUNCIONARIO(
18. MATRICULA INT PRIMARY KEY IDENTITY,
19. NOME VARCHAR (255) NOT NULL,
20. DATANASCIMENTO DATE NOT NULL,
21. CPF CHAR(11) UNIQUE CHECK (LEN(CPF) = 11),
22. SALARIO MONEY NULL,
23. DATAADMISSAO DATE DEFAULT (GETDATE()),
24. DATADEMISSAO DATE NULL,
25. INICIOFERIAS DATE NULL,
26. FIMFERIAS DATE NULL,
27. STATUS\_FUNC VARCHAR (20) CHECK (STATUS\_FUNC IN ('ATIVO','INATIVO','FERIAS','LICENÇA','INSS')),
28. CODDEPTO INT FOREIGN KEY REFERENCES DEPARTAMENTO (CODIGO),
29. CODSUPERVISOR INT FOREIGN KEY REFERENCES FUNCIONARIO (MATRICULA),
30. CODFUNCAO INT FOREIGN KEY REFERENCES FUNCAO (CODIGO))
31. GO
33. INSERT INTO DEPARTAMENTO
34. VALUES ('TECNOLOGIA DA INFORMACAO'),
35. ('RECURSOS HUMANOS'),
36. ('JURIDICO'),('SELEÇÃO'),
37. ('CONTABILIDADE'),
38. ('CONTAS A PAGAR E RECEBER'),
39. ('DEPARTAMENTO PESSOAL')
40. GO
42. INSERT INTO FUNCAO
43. VALUES ('ESTAGIARIO'), ('ANALISTA JR'),('ANALISTA PL'),('ANALISTA SR'),('COORDENADOR'),('GERENTE')
44. GO
46. INSERT INTO FUNCIONARIO (NOME, DATANASCIMENTO,CPF, SALARIO,DATAADMISSAO,DATADEMISSAO,INICIOFERIAS,FIMFERIAS, STATUS\_FUNC, CODDEPTO, CODSUPERVISOR, CODFUNCAO)
47. VALUES
48. ('ANA MARIA', '2000-01-01','12345678911', 1200.55, '2018-05-01',NULL,NULL,NULL, 'ATIVO', 1, 6, 2 ),
49. ('JOSE HENRIQUE', '1998-11-20','12345678912', 2575.55, '2005-09-01','2017-12-01',NULL,NULL, 'INATIVO', 7, NULL, 3 ),
50. ('ANA MARIA', '2002-08-21','12345678913', 950.00, '2019-01-01',NULL,NULL,NULL, 'ATIVO', 6, NULL, 1),
51. ('LUAN FELIX', '1991-09-28','12345678914', 3500.00, '2013-04-01',NULL,NULL,NULL, 'ATIVO', 3,NULL,2),
52. ('FELIPE JOSE DOS SANTOS', '1996-01-11','12345678915', 4000, '2011-05-01','2015-01-29',NULL,NULL, 'INATIVO', 2, NULL, 3),
53. ('MARCELO JOSE', '1980-10-05','12345678916', 7000, '2000-05-01',NULL,'2019-05-01','2019-06-01', 'ATIVO', 1, NULL, 1),
54. ('MARIANA MARIA', '1987-02-08','12345678917', 4500, '2010-01-01',NULL,NULL,NULL, 'INSS', 1, 6, 3 ),
55. ('JULIANA MARIA DOS SANTOS', '2002-01-01','12345678918', 2000, '2017-05-01',NULL,NULL,NULL, 'LICENÇA', 5, NULL, 2 ),
56. ('MARIA ALICIA', '2001-01-01','12345678919', 950, '2018-05-01',NULL,NULL,NULL, 'ATIVO', 1, 6, 1)
57. GO
59. INSERT INTO FUNCIONARIO (NOME, DATANASCIMENTO,CPF, SALARIO,STATUS\_FUNC, CODDEPTO,CODSUPERVISOR,CODFUNCAO)
60. VALUES ('MARIA ALICIA', '2003-09-18','12345678920', 950, 'ATIVO', NULL, NULL,1)

**SCRIPT DE PARA PRATICAR A UNIÃO DE CONSULTAS**

SCRIPT

1. CREATE DATABASE BD\_UNIAO\_CONSULTAS
2. GO
3. USE BD\_UNIAO\_CONSULTAS
4. GO
5. CREATE TABLE A(
6. ID INT NOT NULL,
7. NOME VARCHAR(10) NOT NULL)
9. CREATE TABLE B(
10. ID INT NOT NULL,
11. NOME VARCHAR(10) NOT NULL)
13. INSERT INTO A VALUES (1, 'José'), (2, 'Maria'), (3, 'João')
14. INSERT INTO B VALUES (2, 'Maria'), (3, 'Guilherme'), (4, 'Vitória')

17. --UNION
19. SELECT ID, NOME FROM A
20. UNION
21. SELECT ID, NOME FROM B
23. --UNION ALL
25. SELECT ID , NOME FROM A
26. UNION ALL
27. SELECT ID , NOME FROM B

30. --EXCEPT
32. SELECT ID, NOME FROM A
33. EXCEPT
34. SELECT ID, NOME FROM B
36. --INTERSECT
38. SELECT ID, NOME FROM A
39. INTERSECT
40. SELECT ID, NOME FROM B

**SCRIPT - FUNÇÕES T-SQL**

Script Funções

1. /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* CONVERSÃO \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
2. /\*DATAS DO SISTEMA\*/
4. SELECT GETDATE() as TEST\_GETDATE
5. SELECT CURRENT\_TIMESTAMP as TEST\_CURRENT
6. SELECT SYSDATETIME() as TEST\_SYSUTCDATETIME
8. /\*CONVERSÃO CAST\*/
10. DECLARE @DT DATETIME
11. SET @DT = GETDATE()
13. SELECT CAST(@DT AS DATE) TEST\_CAST
14. SELECT TRY\_CAST(@DT AS tinyint) TEST\_TRYCAST

17. /\*CONVERSÃO CONVERT\*/
19. DECLARE @DT DATETIME
20. SET @DT = GETDATE()
22. SELECT CONVERT(DATE, @DT) TEST\_CONVERT
23. SELECT CONVERT(VARCHAR, @DT,103) TEST\_CONVERT\_DT
24. SELECT TRY\_CONVERT(tinyint, @DT) TEST\_TRYCCONVERT

27. /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* TEXTO \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
28. /\*SUBSTRING\*/
30. DECLARE @TXT VARCHAR(10) = 'ABCDE'
31. SELECT SUBSTRING(@TXT,1,3) AS TEST\_SUBSTRING
33. /\*LEFT E RIGHT\*/
35. DECLARE @TXT VARCHAR(10) = 'ABCDE'
37. SELECT LEFT (@TXT, 2) TEST\_LEFT
38. SELECT RIGHT(@TXT, 2) TEST\_RIGHT
40. /\*LEN\*/
42. DECLARE @TXT VARCHAR(10) = 'ABCDE'
44. SELECT LEN (@TXT) AS TEST\_LEN
46. /\*CHARINDEX\*/
48. DECLARE @TXT VARCHAR(20) = 'CONSULTA BD'
50. SELECT CHARINDEX(' ', @TXT) TEST\_CHARINDEX

53. /\*PATINDEX\*/
55. DECLARE @TXT VARCHAR(50)
56. SET @TXT = 'CONSULTA BD, ONDE ESTUDAMOS SOBRE OS BDS E LINGUAGEM SQL.'
58. SELECT PATINDEX('%BD%', @TXT) AS TEST\_PATINDEX

61. /\*REPLACE\*/
63. DECLARE @TXT VARCHAR(50)
64. SET @TXT = 'APELIDO : PADAWAN'
66. SELECT REPLACE(@TXT, ':', '>>>') TEST\_REPLACE

69. /\*UPPER E LOWER\*/
71. DECLARE @TXT VARCHAR(10) = 't-SQL'
73. SELECT UPPER(@TXT) TEST\_UPPER
74. SELECT LOWER(@TXT) TEST\_LOWER

77. /\*RTRIM, LTRIM, TRIM\*/
79. DECLARE @TXT VARCHAR(20) = ' T-SQL '
81. SELECT RTRIM(@TXT) TEST\_RTRIM
82. SELECT LTRIM(@TXT) TEST\_LTRIM
83. SELECT RTRIM(LTRIM(@TXT)) TEST\_RTRIM\_LTRIM
84. SELECT TRIM (@TXT) TEST\_TRIM

87. /\*CONCATENANDO TEXTOS\*/
89. DECLARE @TXT VARCHAR(20) = 'SQL SERVER'
91. -->>Erro <<
92. SELECT @TXT + 2017 TEST\_PLUS
94. SELECT @TXT + ' ' + CONVERT (VARCHAR (4), 2017) TEST\_PLUS\_CONVERT
96. -->>Modo de concatenação com CONCAT<<
97. SELECT CONCAT(@TXT, ' ' ,2017) TEST\_CONCAT

100. /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* DATA \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

103. /\*DATEPART\*/
105. SELECT
106. DATEPART(YEAR, GETDATE()) AS 'ANO',
107. DATEPART(QUARTER, GETDATE() ) AS 'QUARTO DO ANO',
108. DATEPART(MONTH, GETDATE()) AS 'MÊS',
109. DATEPART(DAYOFYEAR, GETDATE() ) AS 'DIA DO ANO (DE 1 A 365 )',
110. DATEPART(DAY, GETDATE() ) AS 'DIA',
111. DATEPART(WEEK, GETDATE() ) AS 'SEMANA',
112. DATEPART(WEEKDAY, GETDATE() ) AS 'DIA DA SEMANA',
113. DATEPART(HOUR, GETDATE() ) AS 'HORA',
114. DATEPART(MINUTE, GETDATE() ) AS 'MINUTO',
115. DATEPART(SECOND, GETDATE() ) AS 'SEGUNDO',
116. DATEPART(MILLISECOND, GETDATE() ) AS 'MILISEGUNDO',
117. DATEPART(MICROSECOND, GETDATE() ) AS 'MICROSEGUNDO',
118. DATEPART(NANOSECOND, GETDATE() ) AS 'NANOSEGUNDO'

121. /\*DATEDIFF\*/
123. SELECT
124. DATEDIFF(YEAR, '2018-01-01', GETDATE()) AS 'ANO',
125. DATEDIFF(MONTH, '2018-01-01' , GETDATE()) AS 'MÊS',
126. DATEDIFF(DAY, '2018-01-01' , GETDATE()) AS 'DIAS',
127. DATEDIFF(WEEK, '2018-01-01' , GETDATE()) AS 'SEMANA',
128. DATEDIFF(HOUR, '2018-01-01' , GETDATE()) AS 'HORA',
129. DATEDIFF(MINUTE, '2018-01-01' , GETDATE()) AS 'MINUTO'


133. /\*DATEADD\*/
135. SELECT
136. DATEADD(YEAR, 5, getdate() ) AS 'ANO ADICIONADO',
137. DATEADD(YEAR, -5, getdate() ) AS 'ANO SUBTRAIDO',
138. DATEADD(MONTH, 5, getdate() ) AS 'MÊS',
139. DATEADD(DAY, 5, getdate() ) AS 'DIA',
140. DATEADD(WEEK, 5, getdate() ) AS 'SEMANA',
141. DATEADD(HOUR, 5, getdate() ) AS 'HORA'

144. /\*DATENAME\*/
146. SELECT
147. DATENAME(YEAR,GETDATE()) AS 'ANO',
148. DATENAME(MONTH,GETDATE()) AS 'MES',
149. DATENAME(WEEKDAY,GETDATE()) AS 'DIA DA SEMANA'

152. /\*YEAR, MONTH, DAY, EOMONTH\*/
154. SELECT
155. GETDATE() DATA\_SISTEMA,
156. YEAR(GETDATE() ) AS 'ANO',
157. MONTH(GETDATE() ) AS 'MÊS',
158. DAY(GETDATE() ) AS 'DIA',
159. EOMONTH(GETDATE()) AS 'ULTIMO DIA DO MES'

162. /\*ISDATE\*/
164. SELECT ISDATE(GETDATE()) as TEST\_DATE\_VALID
166. SELECT ISDATE('2018-13-01') as TEST\_DATE\_INVALID

169. /\*TRATAMENTO DE NULOS\*/
171. SELECT
172. ISNULL(NULL, 'SQL') AS TEST\_ISNULL,
173. NULLIF('SQL','SQL') AS TEST\_NULLIF,
174. NULLIF('T-SQL','SQL') AS TEST\_NULLIF2,
175. COALESCE(NULL, NULL, 'SQL') AS TEST\_COALESCE,
176. COALESCE(NULL, 'T-SQL', 'SQL') AS TEST\_COALESCE2

<https://consultabd.wordpress.com/2014/11/27/insert-select-inserindo-dados-com-um-select/>

<https://consultabd.wordpress.com/2015/10/22/realizando-backup-completo-full-sql-server/>

<https://consultabd.wordpress.com/2015/10/27/realizando-backup-diferencial-differential-sql-server/>

<https://consultabd.wordpress.com/2015/11/04/realizando-backup-de-logs-de-transacao-sql-server/>

<https://consultabd.wordpress.com/2020/06/24/montando-trilhas-de-certificacoes-microsoft-para-analytics/>

**Simulado MTA**