

# **Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas**

## **Sistemas Gerenciadores de Banco de Dados**

Professor: André Ulisses da Silva  
E-mail: [andre.ulisses@hotmail.com](mailto:andre.ulisses@hotmail.com)



# Indicadores de Aprendizagem



4. Cria Scripts SQL para inserir, consultar, alterar e recuperar dados.
5. Cria Store Procedures e Triggers para resolver problemas.

# SQL - MANIPULAÇÃO DE DADOS



## Funções

# SQL - MANIPULAÇÃO DE DADOS



## Consultando dados – Funções String

ASCII	Retorna o código ASCII de um valor
CHAR	Retorno uma String com o valor informado
CONCAT	Concatena os valores informados
INSTR	Retorna a posição inicial de uma SubString
LCASE	Converte uma String em letras minúsculas
UCASE	Converte uma String em letras maiúsculas
LEFT	Retorna os caracteres a esquerda da String
RIGHT	Retorna os caracteres a direita da String
LOCATE	Retorna a posicao de uma SubString
LTRIM	Retorna uma string sem espaços a esquerda
RTRIM	Retorna uma string sem espaços a direita

# SQL - MANIPULAÇÃO DE DADOS



## Consultando dados – Funções String

SUBSTRING	Retorna uma substring de uma string
REPLACE	Substitui uma substring em uma string
TRIM	Retira os espaços da esquerda e direita
LENGTH	Retorna o tamanho de uma string
REVERSE	Retorna uma string invertida

# SQL - MANIPULAÇÃO DE DADOS



## Consultando dados – Funções String

### ASCII(str)

Retorna o valor do código ASCII do caracter mais a esquerda da string “str”. Retorna 0 se str é uma string vazia.

```
SELECT ASCII('A');
```

---

### CHAR(N, ...)

CHAR() interpreta os argumentos como inteiros e retorna uma string com caracteres dados pelo valor do código ASCII referentes a estes inteiros. Valores NULL são desconsiderados.

```
SELECT CHAR(77,121,83,81,'76');
```

# SQL - MANIPULAÇÃO DE DADOS



## Consultando dados – Funções String

### CONCAT(N, ...)

Retorna a string resultante da concatenação dos argumentos.

```
SELECT CONCAT('My', 'S', 'QL');
```

---

### INSTR(str, substr)

Retorna a posição da primeira ocorrência da substring “substr” na string “str”. É o mesmo que as o LOCATE() com dois argumentos, exceto pelo fato de que os argumentos estão tracados

```
SELECT INSTR('SENAC', 'N');
```

# SQL - MANIPULAÇÃO DE DADOS



## Consultando dados – Funções String

### LCASE(str)

Retorna a string str com todos caracteres alterados para letra minúsculas

```
SELECT LCASE('SENAC');
```

---

### UCASE(str)

Retorna a string str com todos caracteres alterados para letra maiúsculas

```
SELECT UCASE('senac');
```



# SQL - MANIPULAÇÃO DE DADOS



## Consultando dados – Funções String

**LEFT(str, tam)**

Retorna os “tam” caracteres mais a esquerda da string “str”

```
SELECT LEFT('senac', 3);
```

---

**RIGHT(str, tam)**

Retorna os “tam” caracteres mais a direita da string “str”

```
SELECT RIGHT('senac', 3);
```

# SQL - MANIPULAÇÃO DE DADOS



## Consultando dados – Funções String

### LOCATE(substr, str)

A primeira sintaxe retorna a posição da primeira ocorrência da substring “substr” na string “str”.

```
SELECT LOCATE('n', 'senac');
```

---

### SUBSTRING(str, pos, tam)

Retorna a substring com "tam" caracteres da string "str", iniciando da posição "pos".

```
SELECT SUBSTRING('senac', 2, 3);
```

# SQL - MANIPULAÇÃO DE DADOS



## Consultando dados – Funções String

### LTRIM(str)

Retorna a string “str” com caracteres de espaços extras iniciais removidos.

```
SELECT LTRIM('senac');
```

---

### RTRIM(str)

Retorna a string “str” com caracteres de espaços extras finais removidos.

```
SELECT RTRIM('senac');
```

# SQL - MANIPULAÇÃO DE DADOS



## Consultando dados – Funções String

### TRIM(str)

Retorna a string “str” com caracteres de espaços extras iniciais e finais removidos.

```
SELECT TRIM('senac');
```

---

### REPLACE(str, de\_str, para\_str)

Retorna a string str com todas ocorrências da string “dea\_str” substituída pela string “para\_str”.

```
SELECT REPLAC('seMac', 'M', 'n');
```

# SQL - MANIPULAÇÃO DE DADOS



## Consultando dados – Funções String

### LENGTH(str)

Retorna o tamanho da string "str".

```
SELECT LENGTH('senac');
```

---

### REVERSE(str)

Retorna a str invertida.

```
SELECT REVERSE('senac');
```

# SQL - MANIPULAÇÃO DE DADOS



## Consultando dados – Funções Numéricas

+	Adição
-	Subtração
*	Multiplicação
/	Divisão
%	Retorna o resto da divisão
ABS	Retorna o valor absoluto
SIGN	Retorna o sinal de um valor
MOD	Retorna o resto da divisão
FLOOR	Retorna o maior valor inteiro não maior que o valor informado
CEIL	Retorna o menor valor inteiro não menor que o valor informado

# SQL - MANIPULAÇÃO DE DADOS



## Consultando dados – Funções Numéricas

ROUND	Retorna o valor arredondado para o inteiro mais próximo
DIV	Divisão de inteiros
POW	Retorna um valor elevado a uma determinada potencia
SQRT	Retorna a raiz quadrada de um valor
RAND	Retorna um valor ponto flutuante aleatório
LEAST	Retorna o menor valor em uma lista (Min)
GREATEST	Retorna o maior valor em uma lista (Max)
TRUNCATE	Retorna um valor truncado das casas decimais

# SQL - MANIPULAÇÃO DE DADOS



## Consultando dados – Funções Numérica

+

Adição.

```
SELECT 3 + 4;
```

---

-

Subtração.

```
SELECT 3 - 4;
```



# SQL - MANIPULAÇÃO DE DADOS



## Consultando dados – Funções Numérica

\*

Multiplicação.

```
SELECT 3 * 4;
```

---

/

Divisão.

```
SELECT 3 / 4;
```

# SQL - MANIPULAÇÃO DE DADOS



## Consultando dados – Funções Numérica

%

Retorna o resto da divisão.

```
SELECT 3 % 4;
```

---

**ABS**(x)

Retorna o valor Absoluto de "x".

```
SELECT ABS(-4);
```

# SQL - MANIPULAÇÃO DE DADOS



## Consultando dados – Funções Numérica

**SIGN**(X)

Retorna o sinal do número.

```
SELECT SING(-1);
```

---

**MOD**(N,M)

Retorna o resto da divisão entre “N” e “M”.

```
SELECT MOD(29,9);
```

# SQL - MANIPULAÇÃO DE DADOS



## Consultando dados – Funções Numérica

### DIV

Divisão de inteiros.

```
SELECT 25 DIV 5;
```

---

### POW(X, Y)

Retorna o valor de “X” elevado a potência de “Y”.

```
SELECT POW(2, 2);
```

# SQL - MANIPULAÇÃO DE DADOS



## Consultando dados – Funções Numérica

**LEAST**(X, Y, ...)

Com dois ou mais argumentos, retorna o menor (valor-mínimo) argumento.

```
SELECT LEAST (2, 4, 1, 9, 7);
```

---

**GREATEST**(X, Y, ...)

Com dois ou mais argumentos, retorna o maior (valor-máximo) argumento.

```
SELECT GREATEST(2, 4, 1, 9, 7);
```

# SQL - MANIPULAÇÃO DE DADOS



## Consultando dados – Funções Numérica

**RAND()**

Retorna um valor de ponto flutuante aleatório na faixa de 0 a 1.0.

```
SELECT RAND();
```

# SQL - MANIPULAÇÃO DE DADOS



## Consultando dados – Funções Data e Hora

DATE	Separa a data da hora em um DATETIME
TIME	Separa a hora da data em um DATETIME
DAYOFWEEK	Retorna o dia da semana de um DATE
DAY	Retorna o dia de um DATE
MONTH	Retorna o mês de um DATE
WEEK	Retorna a semana de um DATE
hour	Retorna as hora de um TIME
MINUTE	Retorna os minutos de um TIME
SECOND	Retorna os segundos de um TIME

# SQL - MANIPULAÇÃO DE DADOS



## Consultando dados – Funções Data e Hora

DATE_ADD	Realiza operações de soma e subtração de datas
DATEDIFF	Retorna o número de dias entre duas datas
TIMEDIFF	Retorna o número de dias e horas entre duas datas
DATE_FORMAT	Altera o formato de uma data
LAST_DAY	Retorna o último dia do mês de uma data
CURRENT...	Retorna o valor atual de acordo com o tipo escolhido



# SQL - MANIPULAÇÃO DE DADOS



## Consultando dados – Funções Numérica

**DATE**(expr)

Extraí a parte da data da expressão date ou datetime em “expr”.

```
SELECT DATE('2003-12-31 01:02:03');
```

---

**TIME**(expr)

Extraí a parte da hora da expressão time ou datetime em “expr”.

```
SELECT TIME('2003-12-31 01:02:03');
```

# SQL - MANIPULAÇÃO DE DADOS



## Consultando dados – Funções Numérica

**DAYOFWEEK**(data)

Retorna o índice do dia da semana para data (1 = Domingo, 2 = Segunda, ... 7 = Sábado).

```
SELECT DAYOFWEEK('1998-02-03');
```

---

**WEEK**(data)

A função retorna o número da semana para date.

```
SELECT WEEK('1998-02-03');
```

# SQL - MANIPULAÇÃO DE DADOS



## Consultando dados – Funções Numérica

**DAY** (data)

Retorna o dia do mês para data, na faixa de 1 até 31.

```
SELECT DAY('1998-02-03');
```

---

**MONTH**(data)

Retorna o mês para data, na faixa de 1 até 12.

```
SELECT MONTH('1998-02-03');
```

---

**YEAR**(data)

Retorna o ano para data.

```
SELECT YEAR('1998-02-03');
```

# SQL - MANIPULAÇÃO DE DADOS



## Consultando dados – Funções Numérica

**HOUR**(hora)

Retorna a hora para hora.

```
SELECT HOUR('10:05:32');
```

---

**MINUTE**(hora)

Retorna o minuto para hora.

```
SELECT MINUTE('10:05:32');
```

---

**SECOND**(hora)

Retorna o segundo para hora.

```
SELECT SECOND('10:05:32');
```

# SQL - MANIPULAÇÃO DE DADOS



## Consultando dados – Funções Numérica

**DATE\_ADD**(data, INTERVAL tipo expr)

Estas funções realizam operações aritméticas em datas.

tipo do valor	Formarto esperado da expr
SECOND	SECONDS
MINUTE	MINUTES
HOUR	HOURS
DAY	DAYS
MONTH	MONTHS
YEAR	YEARS

# SQL - MANIPULAÇÃO DE DADOS



## Consultando dados – Funções Numérica

**DATE\_ADD**(data, INTERVAL tipo expr)

Estas funções realizam operações aritméticas em datas.

```
SELECT DATE_ADD('1997-12-31 23:59:59', INTERVAL 1 YEAR);
```

```
SELECT DATE_ADD('1997-12-31 23:59:59', INTERVAL 1 MONTH);
```

```
SELECT DATE_ADD('1997-12-31 23:59:59', INTERVAL 1 DAY);
```

```
SELECT DATE_ADD('1997-12-31 23:59:59', INTERVAL 1 HOUR);
```

```
SELECT DATE_ADD('1997-12-31 23:59:59', INTERVAL 1 MINUTE);
```

```
SELECT DATE_ADD('1997-12-31 23:59:59', INTERVAL 1 SECOND);
```

# SQL - MANIPULAÇÃO DE DADOS



## Consultando dados – Funções Numérica

**DATEDIFF**(expr, expr2)

Retorna o número de dias entre a data inicial “expr” e a data final “expr2”.

```
SELECT DATEDIFF('1997-12-31 23:59:59', '1997-12-30');  
TIMEDIF
```

---

**TIMEDIFF**(expr, expr2)

Retorna o tempo entre a hora inicial expr e a hora final expr2.

```
SELECT TIMEDIFF('1997-12-31 23:59:59', '1997-12-30');
```

# SQL - MANIPULAÇÃO DE DADOS



## Consultando dados – Funções Numérica

**DATE\_FORMAT**(data, formato)

Formata o valor de “data” de acordo com a string “formato”.

Formato	Descrição
%M	Nome do mês (January..December)
%W	Nome da semana (Sunday..Saturday)
%D	Dia do mês com sufixo Inglês (0th, 1st, 2nd, 3rd, etc.)
%Y	Ano, numerico, 4 digitos
%y	Ano, numerico, 2 digitos
%X	Ano para a semana onde o Domingo é o primeiro dia da semana, numerico, 4 digitos; usado com %V



# SQL - MANIPULAÇÃO DE DADOS



## Consultando dados – Funções Numérica

**DATE\_FORMAT**(data, formato)

Formata o valor de “data” de acordo com a string “formato”.

Formato	Descrição
%x	Ano para a semana onde a segunda é o primeiro dia da semana, numerico, 4 digitos; usado com %v
%a	Nome da semana abreviado (Sun..Sat)
%d	Dia do mês, numerico (00..31)
%m	Dia do mês, numerico (0..31)
%c	Mês, numerico (0..12)
%b	Nome do mês abreviado (Jan..Dec)

# SQL - MANIPULAÇÃO DE DADOS



## Consultando dados – Funções Numérica

**DATE\_FORMAT**(data, formato)

Formata o valor de “data” de acordo com a string “formato”.

Formato	Descrição
%j	Dia do ano (001..366)
%H	Hora (00..23)
%h	Hora (01..12)
%i	Mês, numerico (0..12)
%r	Tempo, 12-horas (hh:mm:ss seguido por AM ou PM)
%T	Tempo, 24-horas (hh:mm:ss)

# SQL - MANIPULAÇÃO DE DADOS



## Consultando dados – Funções Numérica

**DATE\_FORMAT**(data, formato)

Formata o valor de “data” de acordo com a string “formato”.

Formato	Descrição
%S	Segundos (00..59)
%s	Segundos (00..59)
%f	Microsegundos (000000..999999)
%p	AM ou PM
%w	Dia da semana (0=Domingo..6=Sabado)
%U	Semana(00..53), onde o Domingo é o primeiro dia da semana.

# SQL - MANIPULAÇÃO DE DADOS



## Consultando dados – Funções Numérica

**DATE\_FORMAT**(data, formato)

Formata o valor de “data” de acordo com a string “formato”.

Formato	Descrição
%u	Semana(00..53), onde a Segunda é o primeiro dia da semana.
%V	Semana(01..53), onde o Domingo é o primeiro dia da semana; usado com %X
%v	Semana(01..53), onde a Segunda é o primeiro dia da semana; usado com %x
%%	Um literal ‘%’.

# SQL - MANIPULAÇÃO DE DADOS



## Consultando dados – Funções Numérica

**DATE\_FORMAT**(data, formato)

Formata o valor de “data” de acordo com a string “formato”.

```
SELECT DATE_FORMAT('1997-10-04 22:23:00', '%W %M %Y');  
SELECT DATE_FORMAT('1997-10-04 22:23:00', '%H:%i:%s');  
SELECT DATE_FORMAT('1997-10-04 22:23:00', '%D %y %a %d %m %b %j');  
SELECT DATE_FORMAT('1997-10-04 22:23:00', '%H %k %I %r %T %S %w');  
SELECT DATE_FORMAT('1999-01-01', '%X %V');
```

# SQL - MANIPULAÇÃO DE DADOS



## Consultando dados – Funções Numérica

**LAST\_DAY**(data)

Pega um valor date ou datetime e retorna o valor correspondente para o último dia do mês.

```
SELECT LAST_DAY('2004-01-01 01:01:01');
```

---

**CURDATE**(), **CURRENT\_DATE**, **CURRENT\_DATE**()

Retorna a data atual.

```
SELECT CURDATE();
```

# SQL - MANIPULAÇÃO DE DADOS



## Consultando dados – Funções Numérica

**CURTIME()**, **CURRENT\_TIME**, **CURRENT\_TIME()**

Retorna a hora atual.

.

```
SELECT CURTIME();
```

---

**SYSDATE()**, **CURRENT\_TIMESTAMP**, **LOCALTIME**, **LOCALTIME()**,  
**CURRENT\_TIMESTAMP()**, **LOCALTIMESTAMP**, **NOW()**,  
**LOCALTIMESTAMP()**

Retorna a data e hora atual.

```
SELECT CURRENT_TIMESTAMP();
```

# SQL - MANIPULAÇÃO DE DADOS



## Consultando dados – Funções Conversão

As funções **CAST()** e **CONVERT()** devem ser usada para tomar um valor de um tipo e produzir um valor de outro tipo. As suas sintaxes são as seguintes:

**CAST**(expressão AS tipo)

**CONVERT**(expressão, tipo)

```
SELECT CAST('2000-01-01' AS DATE);
```

```
SELECT CONVERT('2000-01-01', DATE)
```



andre.ulisses@hotmail.com  
**www.sc.senac.br**

