

Bacharelado em Sistemas de Informação

BANCOS DE DADOS

Aula 15

SQL

Funções Numéricas, Funções de Data, Funções de conversão



✓ FUNÇÕES NUMÉRICAS

– Aceitam números como entrada e retornam valores numéricos

- Mais usadas:

ROUND

TRUNC

CEIL

FLOOR

POWER

SQRT

SIGN

ABS

MOD



- ✓ Para os exemplos dados, usaremos a tabela EMP, fornecida com a instalação do Oracle:

```
SQL> descr emp;
```

Name	Null?	Type
-----	-----	-----
EMPNO	NOT NULL	NUMBER (4)
ENAME		VARCHAR2 (10)
JOB		VARCHAR2 (9)
MGR		NUMBER (4)
HIREDATE		DATE
SAL		NUMBER (7, 2)
COMM		NUMBER (7, 2)
DEPTNO		NUMBER (2)

```
SQL> select * from emp;
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	17/12/80	800		20
7499	ALLEN	SALESMAN	7698	20/02/81	1600	300	30
7521	WARD	SALESMAN	7698	22/02/81	1250	500	30
7566	JONES	MANAGER	7839	02/04/81	2975		20
7654	MARTIN	SALESMAN	7698	28/09/81	1250	1400	30
7698	BLAKE	MANAGER	7839	01/05/81	2850		30
7782	CLARK	MANAGER	7839	09/06/81	2450		10
7788	SCOTT	ANALYST	7566	19/04/87	3000		20
7839	KING	PRESIDENT		17/11/81	5000		10
7844	TURNER	SALESMAN	7698	08/09/81	1500	0	30
7876	ADAMS	CLERK	7788	23/05/87	1100		20
7900	JAMES	CLERK	7698	03/12/81	950		30
7902	FORD	ANALYST	7566	03/12/81	3000		20
7934	MILLER	CLERK	7782	23/01/82	1300		10

14 linhas selecionadas.



✓ FUNÇÕES NUMÉRICAS

▶▶ Função ROUND

- **Formato :**

ROUND (col | valor, n)

- ▶▶ Arredonda a coluna expressão ou valor para n casas decimais.
- ▶▶ Se n é omitido, o arredondamento elimina as casas decimais.
- ▶▶ Se n é negativo, os números à esquerda do ponto decimal são arredondados para zero.

✓ FUNÇÕES NUMÉRICAS

► Função ROUND

Exemplo:

```
SQL> SELECT ROUND (45.923, 1) ,  
2          ROUND (45.923) ,  
3          ROUND (45.323, -1) ,  
4          ROUND (42.323, -2) ,  
5          SAL ,  
6          ROUND (SAL/32, 2)  
7 FROM EMP  
8 WHERE DEPTNO = 10  
9 /
```

ROUND (45.923, 1)	ROUND (45.923)	ROUND (45.323, -1)	ROUND (42.323, -2)	SAL	ROUND (SAL/32, 2)
45, 9	46	50	0	2450	76, 56
45, 9	46	50	0	5000	156, 25
45, 9	46	50	0	1300	40, 63



✓ FUNÇÕES NUMÉRICAS

▶▶ Função TRUNC

- **Formato :**

TRUNC (col | valor, n)

- ▶▶ Trunca a coluna expressão ou valor para *n* casas decimais.
- ▶▶ Se *n* é omitido, o truncamento elimina as casas decimais.
- ▶▶ Se *n* é negativo, os números à esquerda do ponto decimal são truncados para zero.

✓ FUNÇÕES NUMÉRICAS

► Função TRUNC

Exemplo:

```
1 SELECT TRUNC (45.923, 1),
2         TRUNC (45.923),
3         TRUNC (45.323, -1),
4         TRUNC (42.323, -2),
5         SAL,
6         TRUNC (SAL/32, 2)
7 FROM EMP
8* WHERE DEPTNO = 10
SQL> /
```

TRUNC (45.923, 1)	TRUNC (45.923)	TRUNC (45.323, -1)	TRUNC (42.323, -2)	SAL	TRUNC (SAL/32, 2)
45, 9	45	40	0	2450	76, 56
45, 9	45	40	0	5000	156, 25
45, 9	45	40	0	1300	40, 62



✓ FUNÇÕES NUMÉRICAS

▶ Função CEIL

- **Formato :**

CEIL (col | valor)

- ▶ **Encontra o menor inteiro maior ou igual à coluna, expressão ou valor.**

✓ FUNÇÕES NUMÉRICAS

► Função CEIL

- Exemplo:

```
SQL> SELECT CEIL (SAL), CEIL(99.9), CEIL(101.76), CEIL(-11.1)
2  FROM EMP
3  WHERE SAL BETWEEN 3000 AND 5000;
```

CEIL(SAL)	CEIL(99.9)	CEIL(101.76)	CEIL(-11.1)
3000	100	102	-11
5000	100	102	-11
3000	100	102	-11

3 linhas selecionadas.

✓ FUNÇÕES NUMÉRICAS

▶▶ Função FLOOR

- **Formato :**

FLOOR (col | valor)

- ▶▶ **Encontra o maior inteiro menor ou igual à coluna, expressão ou valor.**



✓ FUNÇÕES NUMÉRICAS

► Função FLOOR

- **Formato :**

FLOOR (col | valor)

- **Exemplo:**

```
SQL> SELECT FLOOR(SAL), FLOOR(99.9), FLOOR(101.76), FLOOR(-11.1)
2     FROM EMP
3     WHERE SAL BETWEEN 3000 AND 5000;
```

FLOOR(SAL)	FLOOR(99.9)	FLOOR(101.76)	FLOOR(-11.1)
3000	99	101	-12
5000	99	101	-12
3000	99	101	-12

3 linhas selecionadas.



DML - Comandos para a Manipulação de Dados

✓ FUNÇÕES NUMÉRICAS

▶ Função POWER

- **Formato :**

POWER (col | valor, n)

- ▶ Encontra a coluna, expressão ou valor à enésima potência.
- ▶ *n* pode ser negativo, mas deve ser inteiro.

DML - Comandos para a Manipulação de Dados

✓ FUNÇÕES NUMÉRICAS

► Função POWER

- **Formato :**

POWER (col | valor, n)

- **Exemplo:**

```
SQL> SELECT SAL, POWER(SAL,2), POWER(SAL,3), POWER(50,5)
2 FROM EMP
3 WHERE DEPTNO = 10;
```

SAL	POWER(SAL, 2)	POWER(SAL, 3)	POWER(50, 5)
2450	6002500	1,471E+10	312500000
5000	25000000	1,250E+11	312500000
1300	1690000	2,197E+09	312500000

3 linhas selecionadas.



✓ FUNÇÕES NUMÉRICAS

► Função SQRT

- **Formato :**

SQRT (col | valor)

- **Calcula a raiz quadrada da coluna, expressão ou valor, que não podem ser negativos.**



✓ FUNÇÕES NUMÉRICAS

► Função SQRT

- **Formato :**

SQRT (col | valor)

- **Exemplo:**

```
SQL> SELECT SAL, SQRT (SAL), SQRT (40), SQRT (COMM)
      2 FROM EMP
      3 WHERE COMM > 0;
```

SAL	SQRT (SAL)	SQRT (40)	SQRT (COMM)
1600	40	6,3245553	17,320508
1250	35,355339	6,3245553	22,36068
1250	35,355339	6,3245553	37,416574

3 linhas selecionadas.



✓ FUNÇÕES NUMÉRICAS

▶▶ Função SIGN

- **Formato :**

SIGN (col | valor)

- ▶▶ **Retorna -1, 0, 1 se a coluna, expressão ou valor é um número negativo, zero ou positivo.**



✓ FUNÇÕES NUMÉRICAS

► Função SIGN

- **Formato :**

SIGN (col | valor)

- **Exemplo:**

```
SQL> SELECT SAL - COMM, SIGN(SAL-COMM), COMM - SAL, SIGN(COMM-SAL)
2  FROM EMP
3  WHERE COMM > 0;
```

SAL-COMM	SIGN(SAL-COMM)	COMM-SAL	SIGN(COMM-SAL)
1300	1	-1300	-1
750	1	-750	-1
-150	-1	150	1

3 linhas selecionadas.

✓ FUNÇÕES NUMÉRICAS

▶▶ Função ABS

- **Formato :**

ABS (col | valor)

▶▶ **Retorna o valor absoluto da coluna, expressão ou valor.**



✓ FUNÇÕES NUMÉRICAS

► Função ABS

- **Formato :**

ABS (col | valor)

- **Exemplo:**

```
SQL> SELECT SAL, COMM, COMM-SAL, ABS (COMM-SAL) , ABS (-35)
2 FROM EMP
3 WHERE DEPTNO = 30;
```

SAL	COMM	COMM-SAL	ABS (COMM-SAL)	ABS (-35)
1600	300	-1300	1300	35
1250	500	-750	750	35
1250	1400	150	150	35
2850				35
1500	0	-1500	1500	35
950				35

✓ FUNÇÕES NUMÉRICAS

▶▶ Função MOD

- **Formato :**

MOD (valor1 | valor2)

▶▶ **Encontra o resto da divisão de *valor1* por *valor2*.**



✓ FUNÇÕES NUMÉRICAS

▶▶ Função MOD

• Exemplo:

```
SQL> SELECT SAL, COMM, MOD (SAL, COMM) , MOD (100, 40)
      2 FROM EMP
      3 ORDER BY COMM;
```

SAL	COMM	MOD (SAL, COMM)	MOD (100, 40)
1500	0	1500	20
1600	300	100	20
1250	500	250	20
1250	1400	1250	20
800			20
2975			20
950			20
1300			20
3000			20
1100			20
2850			20
2450			20
3000			20
5000			20



✓ FUNÇÕES DE MANIPULAÇÃO DE DATAS

- ▶▶ O Oracle armazena as datas em um formato numérico interno, que pode variar de 1-jan-4712 AC até 31-dez-4712 DC.
- ▶▶ O formato interno representa:
 - Século
 - Ano
 - Mês
 - Dia
 - Horas
 - Minutos
 - Segundos



✓ FUNÇÕES DE MANIPULAÇÃO DE DATAS

▶▶ SYSDATE

- ▶▶ SYSDATE é uma pseudo-coluna que retorna o valor corrente da data e hora.
- ▶▶ É comum selecionar SYSDATE a partir da tabela interna e vazia chamada DUAL.
- ▶▶ Para mostrar a data corrente:

SELECT SYSDATE FROM DUAL;

```
SQL> SELECT SYSDATE FROM DUAL;
```

```
SYSDATE
```

```
-----
```

```
29/09/02
```



✓ FUNÇÕES DE MANIPULAÇÃO DE DATAS

▶▶ OPERAÇÕES ARITMÉTICAS

- ▶▶ É possível realizar cálculos com datas, utilizando operadores aritméticos.
- ▶▶ Operações possíveis:
 - **date + number** – adiciona um número de dias à data, retornando uma data
 - **date – number** – subtrai um número de dias da data, retornando uma data
 - **date – date** – subtrai uma data de outra, retornando um número de dias.
 - **date + number/24** – adiciona um número de horas à data, retornando uma data.



✓ FUNÇÕES DE MANIPULAÇÃO DE DATAS

▶▶ OPERAÇÕES ARITMÉTICAS

- Exemplos:

```
SQL> SELECT HIREDATE, HIREDATE-7, SYSDATE-HIREDATE  
2 FROM EMP  
3 WHERE HIREDATE LIKE '%06%';
```

HIREDATE	HIREDATE	SYSDATE-HIREDATE
-----	-----	-----
09/06/81	02/06/81	7782,4696

✓ FUNÇÕES DE MANIPULAÇÃO DE DATAS

▶▶ Função MONTHS_BETWEEN

- **Formato :**

MONTHS_BETWEEN (data1| data2)

▶▶ Retorna o número de meses dentre *data1* e *data2*, positivo ou negativo.



✓ FUNÇÕES DE MANIPULAÇÃO DE DATAS

```
SQL> SELECT MONTHS_BETWEEN(SYSDATE, HIREDATE) ,  
2          MONTHS_BETWEEN('01-JAN-98', '05-NOV-99')  
3 FROM EMP  
4 WHERE MONTHS_BETWEEN(SYSDATE, HIREDATE) > 59;
```

MONTHS_BETWEEN(SYSDATE, HIREDATE)	MONTHS_BETWEEN('01-JAN-98', '05-NOV-99')
261,40235	-22,12903
259,30557	-22,12903
259,24106	-22,12903
257,88622	-22,12903
252,04751	-22,12903
256,91848	-22,12903
255,66041	-22,12903
185,33783	-22,12903
250,40235	-22,12903
252,69267	-22,12903
184,2088	-22,12903
249,85396	-22,12903
249,85396	-22,12903
248,2088	-22,12903



✓ FUNÇÕES DE MANIPULAÇÃO DE DATAS

▶▶ Função ADD_MONTHS

- **Formato :**

ADD_MONTHS (data, n)

- ▶▶ Adiciona *n* meses à data, sendo que *n* deve ser inteiro e não negativo.



✓ FUNÇÕES DE MANIPULAÇÃO DE DATAS

► Função ADD_MONTHS

- **Formato :**

ADD_MONTHS (data, n)

- **Exemplo:**

```
SQL> SELECT HIREDATE, ADD_MONTHS (HIREDATE, 3) , ADD_MONTHS (HIREDATE, -3)
2 FROM EMP
3 WHERE DEPTNO=20;
```

HIREDATE	ADD_MONT	ADD_MONT
17/12/80	17/03/81	17/09/80
02/04/81	02/07/81	02/01/81
19/04/87	19/07/87	19/01/87
23/05/87	23/08/87	23/02/87
03/12/81	03/03/82	03/09/81



✓ FUNÇÕES DE MANIPULAÇÃO DE DATAS

▶▶ Função NEXT_DAY

- **Formato :**

NEXT_DAY (data1, char)

- ▶▶ Retorna a data do próximo dia da semana especificado, seguinte à *data1*.
- ▶▶ *Char* pode ser um número representando um dia, ou uma *string*.



✓ FUNÇÕES DE MANIPULAÇÃO DE DATAS

▶ Função NEXT_DAY

- **Formato :**

NEXT_DAY (data1, char)

- **Exemplo:**

```
SQL> SELECT HIREDATE, NEXT_DAY (HIREDATE, ' SEXTA-FEIRA' ) , NEXT_DAY (HIREDATE, 6)  
2 FROM EMP  
3 WHERE DEPTNO=10;
```

HIREDATE	NEXT_DAY	NEXT_DAY
-----	-----	-----
09/06/81	12/06/81	12/06/81
17/11/81	20/11/81	20/11/81
23/01/82	29/01/82	29/01/82



✓ FUNÇÕES DE MANIPULAÇÃO DE DATAS

▶ Função LAST_DAY

- **Formato :**

LAST_DAY (data1)

▶ Retorna o último dia do mês da *data1*.

▶ *Char* pode ser um número representando um dia, ou uma *string*.

- **Exemplo:**

```
SQL> SELECT SYSDATE, LAST_DAY (SYSDATE) , HIREDATE, LAST_DAY (HIREDATE) , LAST_DAY ( '15-FEV-98 ' )
2   FROM EMP
3   WHERE DEPTNO=10;
```

SYSDATE	LAST_DAY	HIREDATE	LAST_DAY	LAST_DAY
-----	-----	-----	-----	-----
29/09/02	30/09/02	09/06/81	30/06/81	28/02/98
29/09/02	30/09/02	17/11/81	30/11/81	28/02/98
29/09/02	30/09/02	23/01/82	31/01/82	28/02/98

✓ FUNÇÕES DE MANIPULAÇÃO DE DATAS

► Função ROUND

- **Formatos:**

ROUND (data1)

- Retorna *data1* com a hora definida como 12:00AM (meia-noite).

ROUND (data1, 'MONTH')

- Retorna o primeiro dia do mês contido em *data1* se ela estiver na primeira metade do mês, caso contrário, o primeiro dia do próximo mês.

ROUND (data1, 'YEAR')

- Retorna o primeiro dia do ano contido em *data1* se ela estiver na primeira metade do ano, caso contrário, o primeiro dia do próximo ano.



✓ FUNÇÕES DE MANIPULAÇÃO DE DATAS

► Função ROUND

- Exemplo:

```
SQL> SELECT SYSDATE, ROUND (SYSDATE, 'MONTH' ) , ROUND (SYSDATE, 'YEAR' )  
2 FROM DUAL;
```

```
SYSDATE  ROUND (SY  ROUND (SY  
-----  -----  
29/09/02 01/10/02 01/01/03
```

✓ FUNÇÕES DE MANIPULAÇÃO DE DATAS

► Função TRUNC

- **Formatos:**

TRUNC (data1,'MONTH')

- Retorna o primeiro dia do mês contido em *data1*.

TRUNC (data1,'YEAR')

- Retorna o primeiro dia do ano contido em *data1*.

```
SQL> SELECT SYSDATE, TRUNC (SYSDATE, 'MONTH' ) , TRUNC (SYSDATE, 'YEAR' )
      2 FROM DUAL;
```

SYSDATE	TRUNC (SY	TRUNC (SY
-----	-----	-----
29/09/02	01/09/02	01/01/02



✓ FUNÇÕES DE CONVERSÃO DE DATAS E NÚMEROS

▶▶ Função TO_CHAR

- **Formato :**

TO_CHAR (num | data, ['fmt'])

- ▶▶ Converte o *número* ou *data* para o formato *fmt*.
- ▶▶ Se *fmt* for omitido será usado o formato *default* DD-MON-YY para data ou um *char* para números.
- ▶▶ Os formatos somente afetam a forma como o dado é mostrado, não alterando seu valor interno.



✓ FUNÇÕES DE CONVERSÃO DE DATAS E NÚMEROS

► Função TO_CHAR

- Exemplos:

```
SQL> SELECT TO_CHAR(SYSDATE, 'DAY,DDTH MONTH YYYY')  
2 FROM DUAL;
```

```
TO_CHAR(SYSDATE, 'DAY,DDTHMONTHYYY  
-----  
DOMINGO          ,29TH SETEMBRO 2002
```

```
SQL> SELECT TO_CHAR(SYSDATE, 'HH:MI:SS')  
2 FROM DUAL;
```

```
TO_CHAR(  
-----  
11:42:46
```



✓ FUNÇÕES DE CONVERSÃO DE DATAS E NÚMEROS

▶▶ Função TO_CHAR

- Exemplos:

```
SQL> SELECT TO_CHAR(SAL, '$9,999')  
2 FROM EMP;
```

TO_CHAR

\$800
\$1,600
\$1,250
\$2,975
\$1,250
\$2,850
\$2,450
\$3,000
\$5,000
\$1,500
\$1,100
\$950
\$3,000
\$1,300



✓ FUNÇÕES DE CONVERSÃO DE DATAS E NÚMEROS

• Formatos de datas :

SCC ou CC	– Século; 'S' coloca o prefixo '-' para BC
YYYY ou SYYYY	– Ano; 'S' coloca o prefixo '-' para BC
YYY ou YY ou Y	– Últimos 3,2,1 dígitos do ano
Y, YYY	– Ano com vírgula na posição indicada
SYEAR ou YEAR	– Ano; 'S' coloca o prefixo '-' para BC
BC ou AD	– Indicador BC /AD
B.C. ou A.D.	– Indicador BC /AD
Q	– Trimestre
MM	– Mês
MONTH	– Nome do mês, com 9 caracteres
MON	– Nome do mês, com 3 caracteres
WW ou W	– Semana do ano ou mês
DDD ou DD ou D	– Dia do ano, mês ou semana
DAY	– Nome do dia, com 9 caracteres

✓ FUNÇÕES DE CONVERSÃO DE DATAS E NÚMEROS

• Formatos de datas :

- DY** – Nome do dia, abreviado com 3 caracteres
- J** – Data juliana; número de dias desde 31-dez-4713 AC
- AM ou PM** – Indicador AM/PM
- A.M. ou P.M.** – Indicador AM/PM
- HH ou HH12** – Hora do dia (1-12)
- HH24** – Hora do dia (1-24)
- MI** – Minutos
- SS** – Segundo
- SSSSS** – Segundos depois da meia-noite (0-86399)
- /.** – Pontuação reproduzida no resultado
- “...”** – String reproduzida no resultado
- fm** – Prefixo de MONTH ou DAY para eliminar espaços
- TH** – Sufixo para número ordinal
- SP** – Sufixo para *spelled-out* (soletrar) o número (ex:DDSP)
- SPTH** – Sufixo para *spelled-out* (soletrar) o número (ex:DDSPTH)



✓ FUNÇÕES DE CONVERSÃO DE DATAS E NÚMEROS

• Formatos de números:

Formato	Significado	Exemplo	
9	Posição numérica	999999	1234
0	Mostra zeros à esquerda	099999	001234
\$	Cifrão flutuante	\$999999	\$1234
.	Ponto decimal na posição indicada	999999.99	1234.00
,	Vírgula na posição indicada	999,999	1,234
MI	Sinal MI à direita de negativos	999999MI	1234-
PR	Parênteses em negativos	999999PR	(1234)
EEEE	Notação científica	99.999EEEE	1.234E+03
V	Multiplica por 10^n (n=9s ap''os V)	9999V99	123400
B	Mostra zeros como brancos	B9999.99	1234.00

✓ FUNÇÕES DE CONVERSÃO DE DATAS E NÚMEROS

▶▶ Função TO_NUMBER

- **Formato :**

TO_NUMBER (char)

▶▶ Converte *char* que contém um número para *number*.



✓ FUNÇÕES DE CONVERSÃO DE DATAS E NÚMEROS

▶▶ Função TO_NUMBER

- **Formato :**

TO_NUMBER (char)

- **Exemplo:**

```
SQL> SELECT EMPNO, ENAME, JOB, SAL  
2 FROM EMP  
3 WHERE SAL > TO_NUMBER('1500');
```

EMPNO	ENAME	JOB	SAL
7499	ALLEN	SALESMAN	1600
7566	JONES	MANAGER	2975
7698	BLAKE	MANAGER	2850
7782	CLARK	MANAGER	2450
7788	SCOTT	ANALYST	3000
7839	KING	PRESIDENT	5000
7902	FORD	ANALYST	3000

✓ FUNÇÕES DE CONVERSÃO DE DATAS E NÚMEROS

▶▶ Função TO_DATE

- **Formato :**

TO_DATE (char, 'fmt')

- ▶▶ Converte o valor de *char* representando uma data, em uma data de acordo com o formato *fmt* especificado.
- ▶▶ Se *fmt* for omitido, será usado DD-MON-YY.

✓ FUNÇÕES DE CONVERSÃO DE DATAS E NÚMEROS

► Função TO_DATE

- Exemplo:

```
SQL> SELECT EMPNO, ENAME, HIREDATE
2  FROM EMP
3  WHERE HIREDATE = TO_DATE ('Dezembro 3,1981', 'Month dd,yyyy');
```

EMPNO	ENAME	HIREDATE
7900	JAMES	03/12/81
7902	FORD	03/12/81

✓ FUNÇÕES QUE ACEITAM QUALQUER TIPO DE DADO

▶▶ Função DECODE

- **Formato :**

DECODE (col | expressão, busca1, resultado1,
[busca2, resultado2, ...] default)

- ▶▶ Realiza busca condicionais, funcionando como um *case* ou *if*.
- ▶▶ *Col* / expressão são comparados com cada argumento *busca* e retorna o correspondente *resultado* quando a combinação for verdadeira.
- ▶▶ Se o valor *default* for omitido, será retornado um NULL.

✓ FUNÇÕES QUE ACEITAM QUALQUER TIPO DE DADO

Função DECODE

```
SQL> SELECT ENAME, JOB,
DECODE (JOB, 'CLERK', 'WORKER', 'MANAGER', 'BOSS', 'INDEFINIDO') CARGO
2 FROM EMP;
```

ENAME	JOB	CARGO
-----	-----	-----
SMITH	CLERK	WORKER
ALLEN	SALESMAN	INDEFINIDO
WARD	SALESMAN	INDEFINIDO
JONES	MANAGER	BOSS
MARTIN	SALESMAN	INDEFINIDO
BLAKE	MANAGER	BOSS
CLARK	MANAGER	BOSS
SCOTT	ANALYST	INDEFINIDO
KING	PRESIDENT	INDEFINIDO
TURNER	SALESMAN	INDEFINIDO
ADAMS	CLERK	WORKER
JAMES	CLERK	WORKER
FORD	ANALYST	INDEFINIDO

✓ **FUNÇÕES QUE ACEITAM QUALQUER TIPO DE DADO**

Função DECODE

```
SQL> SELECT JOB, SAL,  
DECODE (JOB, 'ANALYST', SAL*1.1, 'CLERK', SAL*1.15, 'MANAGER', SAL*0.95, SAL) CORRIGIDO  
  
2 FROM EMP;
```

JOB	SAL	CORRIGID
-----	-----	-----
CLERK	800	920
SALESMAN	1600	1600
SALESMAN	1250	1250
MANAGER	2975	2826,25
SALESMAN	1250	1250
MANAGER	2850	2707,5
MANAGER	2450	2327,5
ANALYST	3000	3300
PRESIDENT	5000	5000
SALESMAN	1500	1500
CLERK	1100	1265
CLERK	950	1092,5
ANALYST	3000	3300
CLERK	1300	1495



✓ FUNÇÕES QUE ACEITAM QUALQUER TIPO DE DADO

► Função NVL

- **Formato :**

NVL (col | valor, val)

► Converte um valor NULL para *val*.

- **Exemplo:**

```
SQL> SELECT SAL, COMM, SAL*12+NVL (COMM, 0) , NVL (COMM, 1000) , SAL*12+NVL (COMM, 1000)
2 FROM EMP
3 WHERE DEPTNO =10;
```

SAL	COMM	SAL*12+NVL (COMM, 0)	NVL (COMM, 1000)	SAL*12+NVL (COMM, 1000)
2450		29400	1000	30400
5000		60000	1000	61000
1300		15600	1000	16600



✓ FUNÇÕES QUE ACEITAM QUALQUER TIPO DE DADO

▶▶ Função GREATEST

- **Formato :**

GREATEST (col | valor1, col | valor2, ...)

▶▶ Retorna o maior valor da lista de valores

- **Exemplo:**

```
SQL> SELECT GREATEST(1000,2000) , GREATEST(SAL, COMM)
2   FROM EMP
3   WHERE DEPTNO =30;
```

GREATEST(1000,2000)	GREATEST(SAL, COMM)
-----	-----
2000	1600
2000	1250
2000	1400
2000	
2000	1500
2000	

EACH

✓ FUNÇÕES QUE ACEITAM QUALQUER TIPO DE DADO

▶ Função LEAST

- **Formato :**

LEAST (col | valor1, col | valor2, ...)

▶ Retorna o menor valor da lista de valores

- **Exemplo:**

```
SQL> SELECT LEAST(1000,2000) , LEAST(SAL, COMM)
2 FROM EMP
3 WHERE DEPTNO =30;
```

LEAST(1000,2000)	LEAST(SAL, COMM)
-----	-----
1000	300
1000	500
1000	1250
1000	
1000	0
1000	



✓ FUNÇÕES QUE ACEITAM QUALQUER TIPO DE DADO

▶▶ Função VSIZE

- **Formato :**

VSIZE (col | valor)

▶▶ Retorna o número de bytes na representação interna do Oracle.

- **Exemplo:**

```
SQL> SELECT DEPTNO, VSIZE(DEPTNO), VSIZE(HIREDATE), SAL, VSIZE(SAL), ENAME, VSIZE(ENAME)
2 FROM EMP
3 WHERE DEPTNO =10;
```

DEPTNO	VSIZE(DEPTNO)	VSIZE(HIREDATE)	SAL	VSIZE(SAL)	ENAME	VSIZE(ENAME)
10	2	7	2450	3	CLARK	5
10	2	7	5000	2	KING	4
10	2	7	1300	2	MILLER	6



Exercícios

Considerando a tabela de músicas criada anteriormente, faça as consultas solicitadas.

1. Mostre a raiz quadrada dos tempos de cada música, em segundos.
2. Mostre o tempo de cada música arredondado, em minutos.
3. Mostre o tempo exato de cada música em minutos e segundos no formato mm:ss.
4. Gere uma lista de músicas, da seguinte forma: 'nome – tempo – situação'. Em situação deve ser mostrado “Maior 2 minutos” ou “Menor ou igual” se o tempo de duração for maior, igual ou menor que dois minutos que 2 minutos.



Exercícios

Usando a base de dados de empresas, já criada:

5. Mostre o nome dos funcionários e salários acrescidos de 15% e expressos em valores inteiros.
6. Mostre a raiz quadrada dos salários e o resto da comissão dividido por 13 de todos os funcionários.
7. Mostre o nome, a data de admissão e há quantos dias o funcionário trabalha na empresa para os empregados do departamento 20.
8. Mostre o nome de cada empregado com data de admissão e data de revisão do salário (que é de um ano após a admissão), em ordem ascendente de data de revisão salarial.
10. Mostre os salários dos funcionários se estes forem maior que a comissão ou a comissão, caso seja maior que o salário.



**Bacharelado em
Sistemas de Informação**

BANCOS DE DADOS

Aula 15

SQL

Funções Numéricas, Funções de Data, Funções de conversão

