



Estrutura e Banco de Dados

Utilizando Funções em SQL

Agenda

•Manuseando Valores Nulos

•Funções Alfanuméricas

- Lower
- Upper
- InitCap
- Lpad/Rpad
- Substr
- Instr
- Length
- Translate/Replace

•Funções Numéricas

- Round
- Trunc
- Power
- SQRT
- MOD
- Sum
- AVG
- Count
- Max
- Min

•Funções de Conversão

- To_Char
- Exercícios
- Bibliografia

Pré requisito para este material

Se logar com o usuário HR e Selecionar SQL Commands

ORACLE® Database Express Edition

Database Login

Enter your database username and password.

Username

Password

Login

Links

- [License Agreement](#)
- [Documentation](#)
- [Forum Registration](#)
- [Discussion Forum](#)
- [Product Page](#)

[Click here to learn how to get started](#)

ORACLE® Database Express Edition

User: HR

Home



Administration



Object Browser



SQL



Utilities



Application Builder

Funções são usadas para manipular dados.

Elas aceitam um ou mais argumentos e retorna um valor.

Um argumento é uma constante, refere-se a variável ou coluna.

O formato para uma função é a seguinte:

`função_nome (arumento1,argument2,...)`

Funções podem ser usadas para:

- Cálculos sobre datas;
- Modificar valores de itens individuais;
- Manipular saída para grupos de linhas;
- Alterar formatos de datas para mostrá-los.

Manuseando Valores Nulos

Um valor nulo é um valor indisponível e desconhecido. O valor nulo não é zero. Zero é um número. Nulo é ausência de valor.

Se algum valor de uma coluna em uma expressão for nulo, o resultado será nulo. Na declaração seguinte, a coluna Salário Anual tem um resultado na remuneração

```
SELECT First_Name, Last_Name,  
Salary*12+Commission_pct SALARIO_ANUAL  
FROM Employees;
```

FIRST_NAME	LAST_NAME	SALARIO_ANUAL
Steven	King	-
Neena	Kochhar	-
Lex	De Haan	-
Alexander	Hunold	-
Bruce	Ernst	-
David	Austin	-
Valli	Pataballa	-
Diana	Lorentz	-
Nancy	Greenberg	-
Daniel	Faviet	-
More than 10 rows available. Increase rows selector to view more rows.		

Manuseando Valores Nulos

Para realizar o cálculo para todos os empregados, é necessário converter os valores nulos para numéricos. A função NVL converte valores nulos para não nulos.

```
SELECT First_Name, Last_Name,  
Salary*12+NVL(Commission_pct,0) SALARIO_ANUAL  
FROM Employees;
```

FIRST_NAME	LAST_NAME	SALARIO_ANUAL
Steven	King	288000
Neena	Kochhar	204000
Lex	De Haan	204000
Alexander	Hunold	108000
Bruce	Ernst	72000
David	Austin	57600
Valli	Pataballa	57600
Diana	Lorentz	50400
Nancy	Greenberg	144000
Daniel	Faviet	108000
More than 10 rows available. Increase rows selector to view more rows.		

NVL conta com dois argumentos:

- 1) uma expressão
- 2) um valor não nulo.

Note que você pode usar a função NVL para converter qualquer tipo de valor nulo.

NVL(COLUNA_DATA_NULA,'30-OCT-74')

NVL(COLUNA_NUMÉRICA_NULA,21)

NVL(COLUNA_ALFANUMÉRICA_NULA,'QUALQUER VALOR')

LOWER

Funções Alfanuméricas aceitam dados alfanuméricos e podem retornar alfanumérico ou valores numéricos.

Lower(col/valor) - Converte o campo ou expressão para minúsculo

Para mostrar o nome dos departamentos e a constante BANCO DE DADOS em letra minúscula, faça:

```
SELECT LOWER(department_name), LOWER('BANCO DE DADOS')  
FROM DEPARTMENTS;
```

LOWER(DEPARTMENT_NAME)	LOWER('BANCODEDADOS')
administration	banco de dados
marketing	banco de dados
purchasing	banco de dados
human resources	banco de dados
shipping	banco de dados
it	banco de dados
public relations	banco de dados
sales	banco de dados
executive	banco de dados
finance	banco de dados
More than 10 rows available. Increase rows selector to view more rows.	

```
SELECT First_name, Salary  
FROM employees  
WHERE lower(First_name) = 'steven' ;
```

FIRST_NAME	SALARY
Steven	24000
Steven	2200

UPPER

Upper(col/valor) - Converte o campo ou expressão para maiúsculo

Para mostrar o nome dos Funcionários em maiúsculo, faça:

```
SELECT UPPER(First_Name), Upper>Last_Name)  
FROM EMPLOYEES;
```

UPPER(FIRST_NAME)	UPPER(LAST_NAME)
ELLEN	ABEL
SUNDAR	ANDE
MOZHE	ATKINSON
DAVID	AUSTIN
HERMANN	BAER
SHELLI	BAIDA
AMIT	BANDA
ELIZABETH	BATES
SARAH	BELL
DAVID	BERNSTEIN
More than 10 rows available. Increase rows selector to view more rows.	

```
SELECT First_name, Salary  
FROM employees  
WHERE UPPER(First_name) = 'STEVEN' ;
```

FIRST_NAME	SALARY
Steven	24000
Steven	2200

INITCAP

Initcap(col/valor) - Converte as iniciais de cada palavra para maiúsculo

```
SELECT 'Nome: ' || InitCap(First_Name || ' ' || Last_Name) Nome  
FROM EMPLOYEES;
```

NOME
Nome: Ellen Abel
Nome: Sundar Ande
Nome: Mozhe Atkinson
Nome: David Austin
Nome: Hermann Baer
Nome: Shelli Baida
Nome: Amit Banda
Nome: Elizabeth Bates
Nome: Sarah Bell
Nome: David Bernstein
More than 10 rows available. Increase rows selector to view more rows.

LPAD

LPAD(col/valor,n,'character') - Completa a esquerda com o caracter especificado com tamanho de n posições.

```
SELECT LPAD(Department_name,20,'*'),  
       LPAD(Department_name,20),  
       LPAD(Department_name,20,'.')  
FROM Departments;
```

LPAD(DEPARTMENT_NAME,20,'*')	LPAD(DEPARTMENT_NAME,20)	LPAD(DEPARTMENT_NAME,20,'.')
*****Administration	AdministrationAdministration
*****Marketing	MarketingMarketing
*****Purchasing	PurchasingPurchasing
****Human Resources	Human ResourcesHuman Resources
*****Shipping	ShippingShipping
*****IT	ITIT
***Public Relations	Public RelationsPublic Relations
*****Sales	SalesSales
*****Executive	ExecutiveExecutive
*****Finance	FinanceFinance
More than 10 rows available. Increase rows selector to view more rows.		

RPAD

RPAD(col/valor,n,'character') - Completa a direita com o caracter especificado com tamanho de n posições.

```
SELECT RPAD(Department_name,20,'*'),  
       RPAD(Department_name,20),  
       RPAD(Department_name,20, '.')  
FROM Departments;
```

RPAD(DEPARTMENT_NAME,20,'*')	RPAD(DEPARTMENT_NAME,20)	RPAD(DEPARTMENT_NAME,20, '.')
Administration*****	Administration	Administration.....
Marketing*****	Marketing	Marketing.....
Purchasing*****	Purchasing	Purchasing.....
Human Resources****	Human Resources	Human Resources.....
Shipping*****	Shipping	Shipping.....
IT*****	IT	IT.....
Public Relations***	Public Relations	Public Relations....
Sales*****	Sales	Sales.....
Executive*****	Executive	Executive.....
Finance*****	Finance	Finance.....
More than 10 rows available. Increase rows selector to view more rows.		

SUBSTR

Substr(col/valor,pos,n) - Retorna parte de uma expressão, tendo como parâmetro a posição inicial (pos) e a quantidade de caracteres (n).

```
SELECT First_Name || ' ' || Last_Name Nome,  
       Substr(First_Name || ' ' || Last_Name,2,3) Particionado  
FROM EMPLOYEES;
```

Retorna 3 posições a partir da 2ª posição

NOME	PARTICIONADO
Ellen Abel	lle
Sundar Ande	und
Mozhe Atkinson	ozh
David Austin	avi
Hermann Baer	erm
Shelli Baida	hel
Amit Banda	mit
Elizabeth Bates	liz
Sarah Bell	ara
David Bernstein	avi
More than 10 rows available. Increase rows selector to view more rows.	

Concatenando Funções:

```
SELECT First_Name || ' ' || Last_Name Nome,  
       Substr(Upper(First_Name || ' ' || Last_Name),2,3) Particionado  
FROM EMPLOYEES;
```

INSTR

Instr(col/valor,'character') - Retorna a posição da primeira ocorrência do character.

ou

Instr(col/valor,'character',pos,n) - Retorna a posição da "n" ocorrência do character a partir da posição "pos".

```
SELECT Department_name,  
       Instr(Upper(Department_name),'A'),  
       Instr(Upper(Department_name),'ES'),  
       Instr(Upper(Department_name),'A',1,2)
```

FROM Departments;

DEPARTMENT_NAME	INSTR(UPPER(DEPARTMENT_NAME),'A')	INSTR(UPPER(DEPARTMENT_NAME),'ES')	INSTR(UPPER(DEPARTMENT_NAME),'A',1,2)
Administration	1	0	10
Marketing	2	0	0
Purchasing	6	0	0
Human Resources	4	8	0
Shipping	0	0	0
IT	0	0	0
Public Relations	11	0	0
Sales	2	4	0
Executive	0	0	0
Finance	4	0	0

More than 10 rows available. Increase rows selector to view more rows.

LENGTH

Length(col/valor) - Retorna a quantidade de caracteres que uma expressão ou campo tem.

```
SELECT Department_name,  
       Length(Department_name)  
FROM Departments;
```

DEPARTMENT_NAME	LENGTH(DEPARTMENT_NAME)
Administration	14
Marketing	9
Purchasing	10
Human Resources	15
Shipping	8
IT	2
Public Relations	16
Sales	5
Executive	9
Finance	7
More than 10 rows available. Increase rows selector to view more rows.	

TRANSLATE

Translate(col/valor, de, para) - Substitui o caracter ou expressão "de", para o caracter "para". Se o parâmetro para for omitido o mesmo será removido.

```
SELECT First_Name, TRANSLATE(Upper(First_name), 'L', 'X')  
FROM Employees;
```

FIRST_NAME	TRANSLATE(UPPER(FIRST_NAME), 'L', 'X')
Ellen	EXXEN
Sundar	SUNDAR
Mozhe	MOZHE
David	DAVID
Hermann	HERMANN
Shelli	SHEXXI
Amit	AMIT
Elizabeth	EXIZABETH
Sarah	SARAH
David	DAVID
More than 10 rows available. Increase rows selector to view more rows.	

REPLACE

Replace(col/valor,de,para) - Substitui a expressão "de", para a expressão "para". Se o parâmetro para for omitido o mesmo será removido.

```
SELECT Job_Title,  
       REPLACE(Job_Title,'Finance Manager','Gerente Financeiro'),  
       REPLACE(Job_Title,'nt','XX')  
FROM Jobs;
```

JOB_TITLE	REPLACE(JOB_TITLE,'FINANCEMANAGER','GERENTEFINANCEIRO')	REPLACE(JOB_TITLE,'NT','XX')
President	President	PresideXX
Administration Vice President	Administration Vice President	Administration Vice PresideXX
Administration Assistant	Administration Assistant	Administration AssistaXX
Finance Manager	Gerente Financeiro	Finance Manager
Accountant	Accountant	AccouXXaXX
Accounting Manager	Accounting Manager	AccouXXing Manager
Public Accountant	Public Accountant	Public AccouXXaXX
Sales Manager	Sales Manager	Sales Manager
Sales Representative	Sales Representative	Sales RepreseXXative
Purchasing Manager	Purchasing Manager	Purchasing Manager
More than 10 rows available. Increase rows selector to view more rows.		

A Função REPLACE substitui uma expressão por outra e a TRANSLATE substitui caractere por caractere.

ROUND

Round(col/valor,n) - Arredonda o valor com "n" casas decimais. Se "n" for omitido não será definido casas decimais. Valores de 5 ou superior, arredondará para cima e inferiores a 5 o arredondamento será para baixo.

```
SELECT ROUND(45.923,1), ROUND(45.923), ROUND(45.323,1),  
        ROUND(45.323,-1), Salary/32, ROUND(Salary/32,2)  
FROM Employees;
```

ROUND(45.923,1)	ROUND(45.923)	ROUND(45.323,1)	ROUND(45.323,-1)	SALARY/32	ROUND(SALARY/32,2)
45.9	46	45.3	50	750	750
45.9	46	45.3	50	531.25	531.25
45.9	46	45.3	50	531.25	531.25
45.9	46	45.3	50	281.25	281.25
45.9	46	45.3	50	187.5	187.5
45.9	46	45.3	50	150	150
45.9	46	45.3	50	150	150
45.9	46	45.3	50	131.25	131.25
45.9	46	45.3	50	375	375
45.9	46	45.3	50	281.25	281.25
More than 10 rows available. Increase rows selector to view more rows.					

TRUNC

Trunc(col/valor,n) - Trunca o valor com "n" casas decimais. Se "n" for omitido não será definido casas decimais.

```
SELECT Trunc(45.923,1), Trunc(45.923), Trunc(45.323,1),  
       Trunc (45.323,-1), Salary/32, Trunc(Salary/32,2)  
FROM Employees;
```

TRUNC(45.923,1)	TRUNC(45.923)	TRUNC(45.323,1)	TRUNC(45.323,-1)	SALARY/32	TRUNC(SALARY/32,2)
45.9	45	45.3	40	750	750
45.9	45	45.3	40	531.25	531.25
45.9	45	45.3	40	531.25	531.25
45.9	45	45.3	40	281.25	281.25
45.9	45	45.3	40	187.5	187.5
45.9	45	45.3	40	150	150
45.9	45	45.3	40	150	150
45.9	45	45.3	40	131.25	131.25
45.9	45	45.3	40	375	375
45.9	45	45.3	40	281.25	281.25
More than 10 rows available. Increase rows selector to view more rows.					

POWER

Power(col/valor,n) - Eleva um campo ou valor na potência definida em "n".

```
SELECT Power(Salary,2),Power(Salary,2),Power(50,2)  
FROM Employees  
Where Department_id = 30;
```

POWER(SALARY,2)	POWER(SALARY,2)	POWER(50,2)
121000000	121000000	2500
9610000	9610000	2500
8410000	8410000	2500
7840000	7840000	2500
6760000	6760000	2500
6250000	6250000	2500

SQRT

Sqrt(col/valor) - Retorna a raiz quadrada do campo ou valor informado.

```
SELECT SQRT(Salary),SQRT(65536),SQRT(Commission_pct)
FROM Employees
Where Department_id = 30;
```

SQRT(SALARY)	SQRT(65536)	SQRT(COMMISSION_PCT)
104.880884817015154699145351367993759848	256	-
55.6776436283002192211947129891854952048	256	-
53.851648071345040312507104915403295563	256	-
52.9150262212918118100323150727852085142	256	-
50.9901951359278483002822410902278198956	256	-
50	256	-

MOD

Mod(col/valor,valor2) - Retorna o resto da divisão do campo ou valor pelo "valor2"

```
SELECT Salary,Commission_pct*33,MOD(Salary,Commission_pct*33), MOD(100,40)
FROM Employees
Where Commission_pct <> 0;
```

SALARY	COMMISSION_PCT*33	MOD(SALARY,COMMISSION_PCT*33)	MOD(100,40)
14000	13.2	8	20
13500	9.9	6.3	20
12000	9.9	1.2	20
11000	9.9	1.1	20
10500	6.6	6	20
10000	9.9	1	20
9500	8.25	4.25	20
9000	8.25	7.5	20
8000	6.6	.8	20
7500	6.6	2.4	20
More than 10 rows available. Increase rows selector to view more rows.			

Função	Descrição
SUM	Retorna a soma de N
AVG	Retorna a média aritmética de N
COUNT	Retorna o número de linhas da consulta.
MAX	Retorna o valor máximo de N
MIN	Retorna o valor mínimo de N

```
SELECT sum(Salary),  
       avg(Salary),  
       count(*),  
       min(Salary),  
       max(Salary)  
FROM employees;
```

SUM(SALARY)	AVG(SALARY)	COUNT(*)	MIN(SALARY)	MAX(SALARY)
691400	6461.68224299065420560747663551401869159	107	2100	24000

```
SELECT count(distinct department_id)  
FROM employees;
```

COUNT(DISTINCTDEPARTMENT_ID)
11

```
SELECT count(*)  
FROM employees  
WHERE Salary > 10000;
```

COUNT(*)
15

```
SELECT sum(Salary)  
FROM employees  
WHERE Department_id = 10;
```

SUM(SALARY)
4400

TO_CHAR

TO_CHAR(data, 'mascara') - Especifica que a data está sendo convertida para um novo formato na saída.

A função **TO_CHAR** é freqüentemente usada para modificar um formato de data padrão para um formato alternativo.

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

```
TO_CHAR(SYSDATE, 'DAYDDTHMONTHYYYY')
```

```
MONDAY 01ST JUNE 2009
```

OBS.: SYSDATE retorna a data e hora atual.

TO_CHAR

TO_CHAR pode também ser usado para extrair o horário de um único dia, e mostrá-lo no especificado formato.

Para mostrar o horário de uma respectiva data:

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

TO_CHAR(SYSDATE,'HH:MI:SS')
10:11:06

OBS.:

Campos do tipo DATE, são utilizados para armazenar Data e Hora.

Dual é uma tabela interna somente para testes, somente com 1 campo e 1 registro

Formatos Alfanuméricos

Máscara	Significado
YYYY ou YYY ou YY ou Y	Últimos 4, 3, 2 ou 1 dígito(s) do ano
Q	Um quarto do Ano
MM	Mês
MONTH	nome do mês, espaçamento com brancos do tamanho de 9 caracteres
MON	nome do mês com 3 letras abreviadas
WW	Semana do ano
DDD ou DD	dia do ano ou dia do mês
DAY	nome do dia da semana , espaçado com brancos com 9 caracteres de tamanho
DY	nome do dia da semana com 3 letras abreviadas
AM ou PM	indicador meridiano
HH ou HH12	horas do dia (0-12)
HH24	horas do dia (0-23)
MI	Minutos
SS	segundos
SSSSS	segundos passado meia-noite(0-86399)

```
SELECT First_Name, Hire_date, TO_CHAR(HIRE_DATE, 'DD/MM/YYYY - HH24:MI:SS')  
FROM Employees;
```

```
SELECT First_Name, Hire_date,  
       TO_CHAR(HIRE_DATE, 'DDD - DD/MM/YYYY - HH24:MI:SSSSS')  
FROM Employees;
```

```
SELECT First_Name, Hire_date, TO_CHAR(HIRE_DATE, 'Q - DD/MONTH/YYYY ')  
FROM Employees;
```

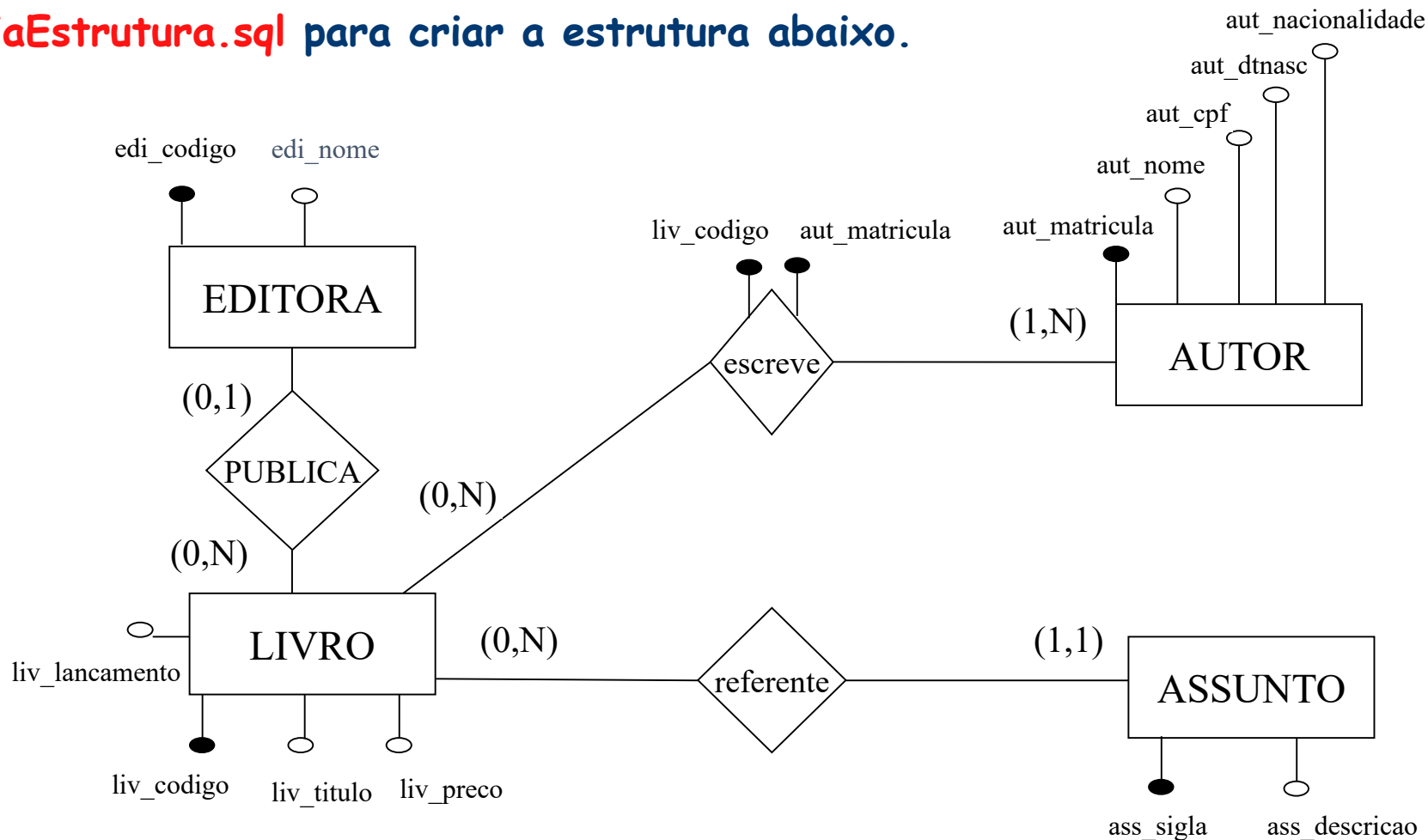
```
SELECT First_Name, Hire_date, TO_CHAR(HIRE_DATE, 'WW - DAY - DD/MONTH/YYYY ')  
FROM Employees;
```

```
SELECT First_Name, Hire_date, TO_CHAR(SYSDATE, 'DD/MM/YYYY') ,  
       TO_CHAR(HIRE_DATE, 'DD/MM/YYYY'),  
       Trunc(Sysdate)-Trunc(Hire_date)  
FROM Employees;
```

OBS.: Neste caso o SQL irá retornar a quantidade de dias entre as duas datas.
Data Atual - A Data de Admissão do funcionário.

Salve os comandos no arquivo **Aula3.sql** e envie via [Link](#).

Com base no der abaixo, e os dados que estão no próximo slide, resolva os exercícios propostos. Caso não tenha criado essas tabelas, utilize o Script **CriaEstrutura.sql** para criar a estrutura abaixo.



LIVRO					
liv_codigo	liv_titulo	liv_preco	liv_lancamento	edi_codigo	ass_sigla
1	banco de dados para web	32,20	10/01/1999	1	BAN
2	programando em linguagem c	30,00	01/10/1997	1	PRO
3	progrmando em linguagem c++	115,50	01/11/1998	3	PRO
4	banco de dados na bioinformática	48,00		2	BAN
5	redes de computadores	42,00	01/09/1996	2	RED

ESCREVE	
liv_codigo	aut_matricula
1	1
2	1
3	2
4	3
5	4

ASSUNTO	
ass_sigla	ass_descricao
BAN	Banco de Dados
PRO	Programação
RED	Redes
SIS	Sistemas Operacionais

EDITORIA	
edi_codigo	edi_nome
1	Mirandela
2	Editora Via Norte
3	Editora Ilhas Tijucas
4	Maria José

AUTOR		
aut_matricula	aut_nome
1	Luiz	
2	Hugo	
3	Joaquim	
4	Regina	

1. Exibir os nomes dos livros, que custam mais de R\$ 100,00 mostrando o as iniciais em maiúsculo e o valor.
2. Mostrar qual é o livro mais caro, mostrando somente o valor.
3. Contar quantas editoras existem.
4. Mostrar o nome do livro, e data de lançamento no formato DD/MM/YYYY dos livros da editora com número 1.
5. Mostrar quantos autores existem no banco de dados.
6. Mostar os nomes dos livros em maiúsculo.
7. Montar a seguinte expressão: "O livro XXX tem YYY caracteres", onde XXX será o nome do livro e YYY a quantidade de caracteres que este livro tem.

Referências Bibliográficas

[1] Fanderuff, Damaris. Dominando o Oracle 9i: Modelagem e desenvolvimento. São Paulo: Pearson Education do Brasil, 2003.

[2] Costa, Rogério Luis de C., SQL : guia prático. 2. ed. Rio de Janeiro : Brasport, 2006.

[3] SILBERSCHATZ, A. Sistema de bancos de dados. São Paulo: Pearson Education do Brasil, 2004.

[4] Morelli, Eduardo M. Terra, 1996. Oracle 9i Fundamental: Sql, Pl/SQL e Administração. São Paulo: Érica, 2002.