


# Disciplina: Banco de Dados



# Funções Oracle

Funções de manipulação de maiúscula e minúscula - **Lower, Upper e Initcap**

- Essas funções (lower, upper, initcap) servem para converter letras maiúsculas em minúsculas e vice-versa.
- 

# Funções Oracle

- **Lower:** Converte caracteres em minúsculas.
- **Upper:** Converte caracteres em maiúsculas.
- **Initcap:** Converte a primeira letra de cada palavra em maiúscula.

```
select upper(first_name), lower(last_name), initcap(job_id), salary from hr.employees
```


upper	lower	initcap	
UPPER(FIRST_NAME)	LOWER(LAST_NAME)	INITCAP(JOB_ID)	SALARY
STEVEN	king	Ad_Pres	24000,00
NEENA	kochhar	Ad_Vp	17000,00
LEX	de haan	Ad_Vp	17000,00
ALEXANDER	hunold	It_Prog	9000,00
BRUCE	ernst	It_Prog	6000,00
DAVID	austin	It_Prog	4800,00
VALLI	pataballa	It_Prog	4800,00
DIANA	lorentz	It_Prog	4200,00



# Funções Oracle

- As funções mencionadas, servem para nos auxiliar também, no filtro das informações a serem buscadas. Na instrução sql abaixo, foi feita uma cláusula where, na coluna last\_name, para buscar um funcionário com o sobrenome 'higgins' (tudo em minúsculo).

```
select e.employee_id, e.first_name, e.last_name from hr.employees e  
where e.last_name = 'higgins'
```



# Funções Oracle

- Não foi encontrada nenhuma linha, pois esse sobrenome está armazenado na tabela como 'Higgins', iniciando com H em maiúsculo. Para solucionar, podemos forçar a saída de todos os dados utilizando uma das funções aprendidas (lower, upper e initcap).

```
select e.employee_id, e.first_name, e.last_name from hr.employees e
where lower(e.last_name) = 'higgins'
```


EMPLOYEE_ID	FIRST_NAME	LAST_NAME
205	Shelley	Higgins

[Download CSV](#)



# Functions de manipulação de caracteres

Function	Resultado
<b>CONCAT</b> ( 'Hello', 'World')	HelloWorld
<b>SUBSTR</b> ('HelloWorld',1,5)	Hello
<b>LENGTH</b> ('HelloWorld')	10
<b>INSTR</b> ('HelloWorld','W')	6
<b>LPAD</b> (salary,10,'*')	*****24000
<b>RPAD</b> (salary,10,'*')	24000*****
<b>REPLACE</b> ('JACK and JUE','J','BL')	BLACK and BLUE
<b>TRIM</b> ('H' FROM 'HelloWorld')	elloWorld





# Functions de manipulação de caracteres

**CONCAT:** Une valores (limitação de dois parâmetros).

**SUBSTR:** Extrai uma string de tamanho determinado.

**LENGHT:** Mostra o tamanho de uma string como um valor numérico.

**INSTR:** Localiza a posição numérica de um caractere nomeado.

**LPAD:** Preenche o valor do caractere à direita.

**RPAD:** Preenche o valor do caractere à esquerda.

**TRIM:** Reduz os caracteres à direita ou à esquerda de uma string de caracteres.



# Functions de manipulação de caracteres

```
select e.employee_id, CONCAT(e.first_name, e.last_name) "NOME", e.job_id,  
LENGTH(e.last_name), INSTR(e.last_name, 'a') "Possui a"  
from hr.employees e  
where SUBSTR(e.job_id, 4) = 'REP'
```

EMPLOYEE_ID	NOME	JOB_ID	LENGTH(E.LAST_NAME)	Possui a
150	PeterTucker	SA_REP	6	0
151	DavidBernstein	SA_REP	9	0
152	PeterHall	SA_REP	4	2
153	ChristopherOlsen	SA_REP	5	0
154	NanetteCambrault	SA_REP	9	2
155	OliverTuvault	SA_REP	7	4
156	JanetteKing	SA_REP	4	0
157	PatrickSully	SA_REP	5	0
158	AllanMcEwen	SA_REP	6	0





# Functions de número

- As functions de número aceitam a entrada numérica e retornam valores numéricos. São utilizadas para tratamento de números e personalização de retorno de valores.

Function	Resultado
<b>ROUND</b> (45.926, 2)	45.93
<b>TRUNC</b> (45.926, 2)	45.92
<b>MOD</b> (1600, 300)	100

- **ROUND:** Arredonda a coluna, expressão ou valor para N casas decimais: se N for omitido, não haverá casas decimais.
- **TRUNC:** Trunca a coluna, expressão ou o valor para N casas decimais: se N for omitido, assumirá o valor default zero.
- **MOD (m,n):** Retorna o restante de m dividido por n.



# Exemplos

## ROUND

```
select ROUND(45.293,2), ROUND(45.293,0) from dual;
```

ROUND(45.293,2)	ROUND(45.293,0)
45,29	45

## TRUNC

```
select TRUNC(45.293,2), TRUNC(45.293,0) from dual;
```

TRUNC(45.293,2)	TRUNC(45.293,0)
45,29	45

# Exemplos

## MOD

```
select last_name, salary, MOD(salary, 5000)
from employees
where job_id='SA_REP';
```

LAST_NAME	SALARY	MOD(SALARY,5000)
King	10000,00	0
Bloom	10000,00	0
Tucker	10000,00	0
Vishney	10500,00	500
Abel	11000,00	1000
Kumar	6100,00	1100
Banda	6200,00	1200
Johnson	6200,00	1200



# Exercícios


1. O departamento de recursos humanos precisa de um relatório para exibir o número do funcionário, o sobrenome, o salário e o salário com 15,5% de aumento (especificado como número inteiro) de cada funcionário. Atribua o apelido `NOVO_SALARIO` para a coluna.
2. Modifique a consulta anterior para adicionar uma coluna que subtraia o salário antigo do novo salário. Atribua o apelido `AUMENTO` à coluna.
3. Crie uma consulta que exiba o sobrenome (com a primeira letra maiúscula e todas as outras minúsculas) e o tamanho do sobrenome de todos os funcionários cujos nomes comecem com a letra J, A ou M. Atribua um apelido apropriado a cada coluna. Classifique os resultados pelos sobrenomes.





# Criação de tabela no banco de dados

Com o comando "CREATE TABLE" é possível criar tabelas no banco de dados. Quando criamos uma tabela, é importante especificar e entender os tipos de dados que estaremos criando, ou seja, no momento da inserção dos dados, os requisitos do tipo de dados devem ser atendidos.






# Criação de tabela no banco de dados

No momento da criação da tabela, é necessário especificar um nome para ela, os tipos de dados das colunas e os respectivos tamanho. Os tipos de dados mais utilizados são:

VARCHAR2(tamanho)	Dados com tamanho variável de caracteres ( tamanho máximo = 4000 )
CHAR(tamanho)	Dados com número de caracteres definido por tamanho
NUMBER( precisão, escala )	Valores numéricos onde: Precisão é o número de caracteres e Escala é o número de caracteres após a virgual ( ou ponto )
DATE	Valores de datas e horários






# Exemplo

Segue exemplo de criação de tabela no banco de dados:  
create table nome\_tabela (coluna tipo de dado (tamanho));

```
1 create table funcionario (cd_func number(20), nome varchar2(30), sobrenome varchar2(30), salario number(30));  
2  
3
```





# Constraints

As constraints impõem regras no nível de tabela, elas por exemplo, impedem a deleção de uma tabela quando existem dependências. Os seguintes níveis de constraints são válidos:

NOT NULL

UNIQUE

PRIMARY KEY

FOREIGN KEY

CHECK







# Primary Key

Exemplo da criação de uma tabela com chave primária:

```
1 create table aluno (cd_aluno number(20) primary key, nome varchar2(30), sobrenome varchar2(30), ds_endereco varchar2(30));  
2  
3
```

*- Lembrar das características da chave primária*






# Foreign Key

Quando utilizamos uma foreign key, precisamos aponta-la para uma outra tabela, fazendo com que ambas se relacionem.

**Foreign Key** – Define a coluna da tabela filha no nível de constraint da tabela.

**References** – Identifica a tabela e a coluna da tabela mãe.

```
create table matricula (cd_matricula number(20) primary key, nome_curso varchar2(30), cd_aluno number(20) references aluno(cd_aluno));
```





# Violando Constraints

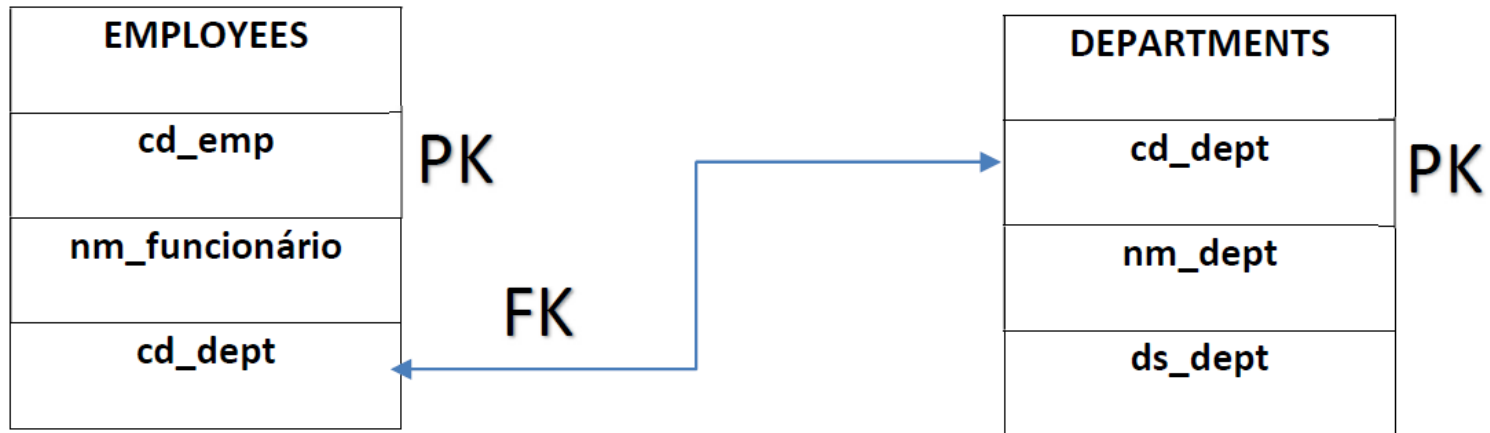
Quando temos relacionamentos entre as tabelas, precisamos nos atentar quanto ao relacionamento entre elas, por exemplo, quando tentamos alterar um registro e esquecemos que a tabela mãe, referente ao registro não, violamos a constraint.

```
UPDATE hr.employees  
SET department_id=55  
WHERE department_id=110;
```

```
UPDATE employees  
*  
ERROR at line 1:  
ORA-02291: integrity constraint (HR.EMP_DEPT_FK)  
violated - parent key not found
```

# Violando Constraints

Na imagem, podemos verificar que para conseguirmos inserir um `cd_dept` na tabela `employees`, esse registro deve primeiramente existir na tabela `departments`, ou seja, o registro mãe.





# Drop Table

O comando drop table é utilizado para excluirmos tabelas dos bancos de dados. Trata-se de um comando simples e que não deixa a possibilidade de desfazer a operação (a não ser por intervenção de um DBA). Segue a sintaxe para execução:

```
DROP TABLE nome_tabela;
```

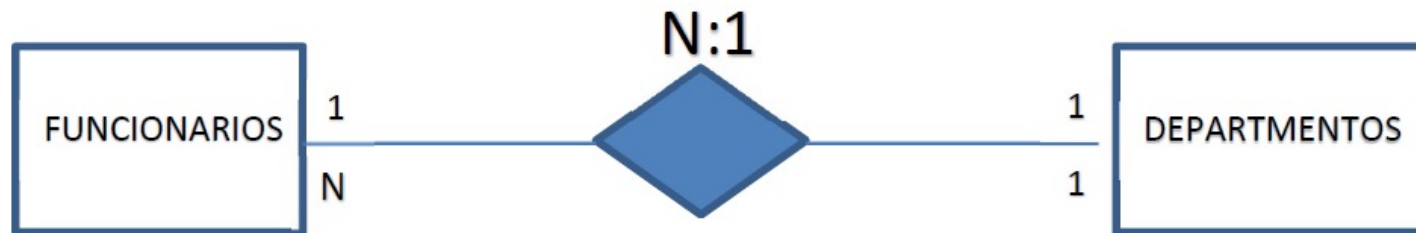
```
6 drop table funcionarios;
```

```
7
```



# Exercícios

4. Criar uma tabela de funcionários e uma tabela de departamentos, utilizando como referência a modelagem de dados abaixo:



## ATRIBUTOS

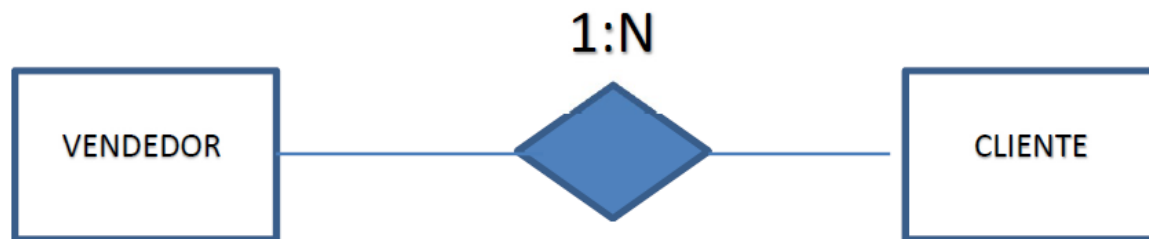
FUNCIONARIOS -> cd\_funcionario, nome, sobrenome, salário (PK cd\_funcionario, FK departamentos (cd\_dept))

DEPARTAMENTOS -> cd\_dept, nome\_dept, ds\_dept (PK cd\_dept)

- a) Inserir 5 registros na tabela funcionários. Qual foi o problema encontrado? Qual a melhor ordem de executar essas inserções?
- b) Inserir 5 registros em cada tabela.

# Exercícios

5. Criar uma tabela de vendedor e cliente, utilizando o modelo entidade relacionamento como:



Atributos

Vendedor -> cd\_vendedor, nm\_vendedor, ds\_endereco


Cliente -> cd\_cliente, nm\_cliente, ds\_endereco, vl\_fatura, vl\_limite

a) Inserir 5 registros em cada tabela.



# Exercícios

6. Excluir todas as tabelas dos exercícios anteriores.







# Universidade **Cruzeiro do Sul**

[www.cruzeirodosul.edu.br](http://www.cruzeirodosul.edu.br)